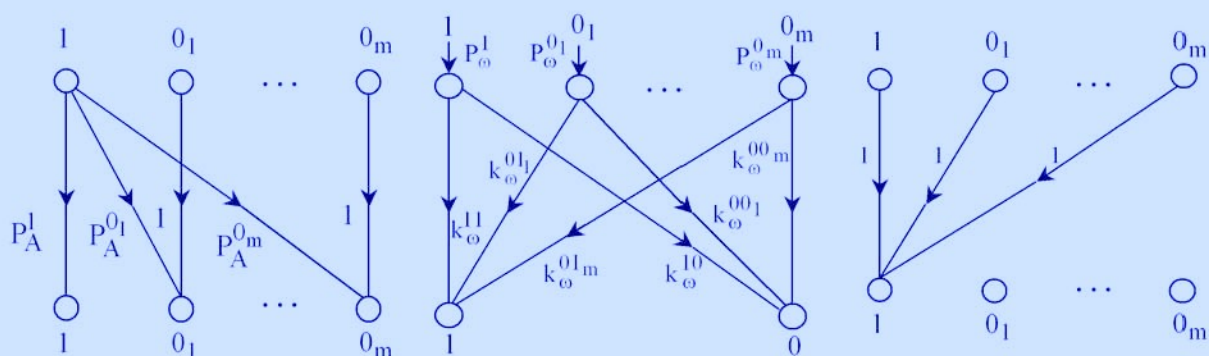


О. П. РОТШТЕЙН  
С. Д. ШТОВБА  
О. М. КОЗАЧКО

---

# МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ НАДІЙНОСТІ БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

---



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

О.П. РОТШТЕЙН

С.Д. ШТОВБА

О.М. КОЗАЧКО

**МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ  
НАДІЙНОСТІ БАГАТОВИМІРНИХ  
АЛГОРИТМІЧНИХ ПРОЦЕСІВ**

**Монографія**

УНІВЕРСУМ-Вінниця  
2007

УДК 681.3

Р 79

*Рецензенти:*

**В. М. Лисогор**, доктор технічних наук, професор

**Р. Н. Квстний**, доктор технічних наук, професор

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 7 від 02.07.07).

**Ротштейн О. П., Штовба С. Д., Козачко О. М.**

Р 79 Моделювання та оптимізація надійності багатовимірних алгоритмічних процесів. – Вінниця: «УНІВЕРСУМ-Вінниця», 2007. – 211 с.

ISBN 978-966-641-231-0

В монографії досліджується моделювання та оптимізація надійності багатовимірних алгоритмічних процесів, при виконанні яких вносяться, виявляються та усуваються помилки різних типів. Пропонуються постановки та методи розв'язання задач чіткої та нечіткої оптимізації надійності таких процесів. Узагальнено багатовимірні моделі надійності операторів, логічних умов та типових алгоритмічних структур на випадок нечітких даних.

Монографія буде корисною проектувальникам, науковим співробітникам, аспірантам та студентам старших курсів технічних університетів, які займаються моделюванням та забезпеченням надійності складних систем.

**УДК 681.3**

ISBN 978-966-641-231-0

© О. П. Ротштейн, С. Д. Штовба, О. М. Козачко, 2007

## ЗМІСТ

<b>Вступ</b>	6
<b>1. Проблема надійності алгоритмічних процесів</b>	9
1.1. Алгоритмічний процес як об'єкт моделювання надійності...	9
1.2. Методи оцінювання та забезпечення надійності алгоритмічних процесів	14
1.3. Принципи моделювання надійності алгоритмічних процесів	29
<b>2. Моделювання надійності багатовимірних алгоритмічних процесів</b>	31
2.1. Аналіз помилок різних типів	31
2.2. Моделі надійності операторів та логічних умов	33
2.3. Моделі надійності алгоритмічних структур	38
2.4. Алгоритм моделювання надійності	41
<b>3. Формалізація задач оптимізації надійності алгоритмічних процесів</b>	42
3.1 Формалізація керованих змінних для задач оптимізації надійності	42
3.2. Показники надійності, які враховуються при оптимізації	44
3.3. Задачі багатокритеріальної оптимізації надійності	44
3.4. Скаляризація задачі багатокритеріальної оптимізації надійності	45
3.4.1. Застосування функції ризику	45
3.4.2. Застосування цільового програмування	46
3.4.3. Застосування умовної оптимізації	47
3.4.4. Спільне застосування умовної оптимізації та цільового програмування	53
<b>4. Градієнтна оптимізація контролів в багатовимірних алгоритмічних процесах</b>	54
4.1. Задачі розподілу контрольних точок	54
4.2. Задачі вибору кратностей контролів	56
4.3. Бібліотека тестових задач	58
4.4. Градієнтна оптимізація надійності багатовимірних алгоритмічних процесів	61
4.4.1. Ідея градієнтного пошуку	61
4.4.2. Градієнт контролю при багатовимірному алгоритмічному процесі	61
4.4.3. Градієнтний розподіл контрольних точок при обмеженнях на безпомилковість	63
4.4.4. Градієнтний розподіл контрольних точок при обмежених ресурсах	67

4.4.5. Градієнтний вибір кратностей контролів при обмеженнях на безпомилковість .....	71
4.4.6. Градієнтний вибір кратностей контролів при обмежених ресурсах .....	73
<b>5. Генетична оптимізація надійності багатовимірних алгоритмічних процесів .....</b>	<b>77</b>
5.1. Ідея генетичного пошуку .....	77
5.2. Генетичне кодування варіанта алгоритмічного процесу .....	78
5.2.1. Задача оптимізації контролів .....	78
5.2.2. Задача вибору варіантів реалізацій операторів та логічних умов .....	79
5.2.3. Загальні задачі структурної оптимізації .....	80
5.2.4. Задачі параметричної оптимізації .....	82
5.2.5. Задачі структурно-параметричної оптимізації .....	82
5.3. Мутація та схрещення хромосом .....	84
5.4. Фітнес-функція хромосом та селекція нової популяції .....	86
5.5. Гібридна генетична оптимізація надійності .....	89
5.5.1. Генерування якісної початкової популяції за ідеями пожадливих методів дискретної оптимізації .....	89
5.5.1.1. Основна ідея .....	89
5.5.1.2. Генерування якісної початкової популяції при виборі способів реалізації операторів та логічних умов .....	90
5.5.1.3. Генерування якісної початкової популяції при оптимізації контролю .....	91
5.5.2. Покращення популяції хромосом мурашиними алгоритмами за острівною схемою .....	92
5.5.2.1. Основна ідея .....	92
5.5.2.2. Графове представлення задач структурної оптимізації .....	94
5.5.2.3. Реалізація основних елементів мурашиного алгоритму .....	95
5.5.2.4. Мурашиний алгоритм оптимізації надійності .....	97
5.5.3. Покращення хромосоми методами локального пошуку .....	98
5.5.3.1. Основна ідея .....	98
5.5.3.2. Застосування градієнтних методів оптимізації контролю .....	98
5.5.3.3. Застосування мурашиних алгоритмів дискретної оптимізації .....	99
5.5.3.4. Застосування класичних методів неперервної оптимізації .....	99

5.5.3.5. Спільне застосування різних методів локального пошуку .....	99
5.6. Експериментальні дослідження з генетичної оптимізації багатовимірних алгоритмічних процесів .....	100
5.6.1. Завдання та методика комп'ютерних експериментів ..	100
5.6.2. Генетичний розподіл контрольних точок при обмеженнях на безпомилковість .....	101
5.6.3. Генетичний розподіл контрольних точок при обмежених ресурсах .....	107
5.6.4. Генетичний вибір кратностей контролів при обмеженнях на безпомилковість .....	110
5.6.5. Генетичний вибір кратностей контролів при обмежених ресурсах .....	115
5.6.6. Порівняння алгоритмів оптимізації надійності багатовимірних алгоритмічних процесів .....	117
5.6.7. Дослідження вірогідності методів оптимізації надійності багатовимірних алгоритмічних процесів ..	122
<b>6. Нечіткі моделі надійності багатовимірних алгоритмічних процесів .....</b>	<b>127</b>
6.1. Нечіткі початкові дані .....	127
6.2. Нечіткі моделі надійності операторів і логічних умов .....	129
6.3. Принцип нечіткого узагальнення .....	130
6.4. Нечіткі моделі надійності типових алгоритмічних структур	132
6.5. Алгоритм аналізу надійності алгоритмічних процесів при нечітких початкових даних .....	149
6.6. Моделювання надійності системи «клієнт-банк» .....	157
<b>7. Нечітка оптимізація надійності алгоритмічних процесів .....</b>	<b>164</b>
7.1. Концепція нечіткої оптимізації надійності .....	164
7.2. Оптимізації за нечіткої мети .....	167
7.3. Оптимізація за нечітких початкових даних .....	172
7.4. Оптимізації за нечіткої мети та нечітких початкових даних .	179
7.5. Особливості генетичного алгоритму нечіткої оптимізації ....	183
<b>Післямова .....</b>	<b>185</b>
<b>Література .....</b>	<b>186</b>
Додаток А. Опис реквізитів форми платіжного доручення системи «клієнт-банк» .....	199
Додаток Б. Класифікація помилок в системі «клієнт-банк». ....	200
Додаток В. Початкові дані для прогнозування часу проведення платіжних доручень в системі «клієнт-банк» .....	203
Додаток Г. Покроковий аналіз надійності системи «клієнт-банк» ..	206

## ВСТУП

В теорії надійності складних систем склалися два принципово різних підходи:

- елементний *S*-підхід, моделі надійності якого побудовані на основі структури системи та характеристик надійності її елементів;
- функціональний *F*-підхід, моделі надійності якого побудовані на основі структури функцій, які виконує система, тобто на основі алгоритму функціонування.

*S*-підхід відповідає класичній теорії надійності систем, в якій за основні показники обрано ймовірність безвідмовної роботи та коефіцієнт готовності. Як початкові дані для моделювання надійності використовуються ймовірнісно-часові характеристики відмов – випадкових подій втрати роботоздатності.

*F*-підхід відповідає теорії надійності алгоритмів, теорії надійності людино-машинних систем, теорії надійності трудових та технологічних процесів. В *F*-підході узагальненим показником надійності виступає ймовірність досягнення мети, яка для прикладних задач інтерпретується показниками безпомилковості, бездефектності, достовірності, своєчасності тощо. Початковими даними для моделей *F*-надійності є ймовірнісно-часові характеристики помилкового виконання окремих функцій – операцій алгоритму функціонування. В *F*-надійності під функціональною відмовою розуміється випадкова подія порушення правильного виконання функції елементом системи за умов збереження їм роботоздатності.

Кожна операція алгоритму реалізуються певними засобами, надійність яких визначається за *S*-підходом. Тому найбільш повним описом надійності складних систем є поєднання *F*- та *S*-підходів. Воно дозволяє покращити надійність, використовуючи як прийоми *S*-підходу (наприклад, резервування засобів, знарядь праці та суб'єктів), так і прийоми *F*-підходу (наприклад, введення в процес функціонування операцій контролю).

Методи покращення *S*-надійності добре розроблені як в теоретичному, так і в практичному (інженерному) зрізах. Теорії надійності алгоритмічних процесів приділено значно менше досліджень.

Під алгоритмічним процесом (АП) будемо розуміти розгорнуту в часі послідовність дій, операцій або робіт, виконання яких забезпечує досягнення мети, тобто отримання кінцевого результату: інформації, знань, документації, продукції тощо. Прикладами АП можуть бути процеси функціонування АСУ та комп'ютерних мереж, навчальні

процеси, процеси виконання науково-дослідних і конструкторських робіт, технологічні процеси тощо.

При проектуванні АП важливо вміти розв'язувати задачі оцінювання та забезпечення таких показників надійності як:

- *ймовірність правильного виконання АП*, яка може інтерпретуватися як достовірність інформації, бездефектність продукції, надійність функціонування алгоритму і т.і.;
- *час виконання АП*, який може використовуватися для оцінювання продуктивності АП або своєчасності досягнення мети.

В більшості робіт моделювання та оптимізація надійності АП здійснюється за бінарною концепцією врахування помилок, де розрізняються лише два стани виконання АП: з помилками або без помилок. Між собою помилки не розрізняються, тобто не важливо, яка саме помилка зроблена. В багатьох реальних задачах використання бінарної концепції врахування помилок є недоцільним, оскільки для різних типів помилок різняться ймовірності їх внесення, виявлення та усунення, так само як і витрати на ці процедури.

В монографії розглядаються так звані *багатовимірні АП*, тобто процеси при виконанні яких вносяться, виявляються та усуваються помилки різних типів. Помилки різних типів, що виникають в АП, пов'язані з неправильним виконанням людиною або технікою деякого фрагменту алгоритму, тобто послідовності операцій, дій або робіт.

Монографія складається з семи розділів.

*В першому розділі* розглядаються багатовимірні АП, аналізуються методи оцінювання та забезпечення їх надійності, пропонуються принципи моделювання та оптимізації надійності таких АП.

*В другому розділі* розглядаються моделі надійності операторів, логічних умов та алгоритмічних структур за багатоарної концепції врахування помилок функціонування.

*В третьому розділі* формулюються постановки задач оптимізації АП, що враховують різноманітні способи управління надійністю. Пропонується загальна постановка задачі векторної оптимізації АП та аналізуються способи її скаляризації за допомогою функції ризику, цільового програмування, умовної оптимізації та їх поєднання. Пропонується підхід до синтезу з загальної задачі оптимізації окремих, які віддзеркалюють практичні потреби управління надійністю АП.

*В четвертому розділі* пропонуються градієнтні алгоритми оптимізації найбільш поширених задача забезпечення надійності багатовимірних АП, а саме розстановки контрольних точок та вибору кратностей контролів.

*В п'ятому розділі* розроблюються реалізації генетичних операцій та схеми гібридного ГА, які необхідні для швидкого знаходження



розв'язків задач оптимізації надійності АП. Перевірка запропонованих моделей та алгоритмів здійснюється на множині тестових задач.

*В шостому розділі* узагальнюються багатовимірні моделі надійності операторів, логічних умов та типових алгоритмічних структур на випадок нечітких початкових даних. Застосування моделей ілюструється на прикладі прогнозування надійності АП проведення платіжного доручення в інтернетівській системі «клієнт-банк».

*В сьомому розділі* пропонуються постановки та методи розв'язання задач забезпечення надійності багатовимірних АП для випадків, коли нечіткими числами задано характеристики надійності операторів і логічних умов та мету і обмеження оптимізації. Нечітка оптимізація здійснюється генетичним алгоритмом, в якому хромосоми оцінюються за принципом Беллмана–Заде з нечіткої теорії прийняття рішень та обираються в нову популяцію за схемою турнірної селекції.

Монографія розвиває ідеї моделювання та оптимізації надійності АП з авторських книг [72, 79, 82]. Роботу над нею автори розподілили таким чином: О.П. Ротштейн написав розділ 2; С.Д. Штовба – розділи 3 і 7 та підрозділи 5.1–5.5; О.М. Козачко – підрозділи 4.4, 5.6 та 6.4–6.6. Решта матеріалу написана авторами спільно. Загальне редагування здійснив О.П.Ротштейн. В монографії використовуються результати кандидатської дисертації О.М.Козачка.

# 1. ПРОБЛЕМА НАДІЙНОСТІ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

Розглядаються багатовимірні АП, аналізуються методи оцінювання та забезпечення їх надійності, пропонуються принципи моделювання та оптимізації надійності таких АП.

## 1.1. Алгоритмічний процес як об'єкт моделювання надійності

Під *алгоритмічним процесом* (АП) розуміється розгорнута у часі послідовність дій, операцій або робіт, виконання яких забезпечує досягнення мети [73]. Цю послідовність дій можна формалізувати деяким алгоритмом з такими властивостями [48]:

- його правильне виконання забезпечує досягнення наперед заданої мети, тобто на виході отримується певний кінцевий результат – продукт праці, інформація, документація, знання тощо;
- він складається з елементарних операцій, які утворюють кінцеву множину;
- послідовність операцій алгоритму детермінована;
- одночасно може виконуватися дискретна, наперед визначена кількість операцій, причому для систем без паралелізму вона дорівнює 1;
- він має початок і кінець, причому перехід з початкового стану в кінцевий здійснюється за кінцеву кількість операцій.

Типовими представниками АП є процеси функціонування АСУ та комп'ютерних мереж, процеси автоматичної та автоматизованої обробки інформації і прийняття рішень, технологічні процеси виробництва продукції, процеси функціонування людино-машинних систем, інженерне проектування, процеси виконання науково-дослідних робіт, процеси навчання, алгоритми діяльності операторів тощо. Класифікація АП представлена на рис. 1.1.

За структурованістю АП поділяють на регулярні та нерегулярні. В регулярних АП відсутні перехрещені зворотні зв'язки, а в нерегулярних – наявні. Надалі розглядатимемо регулярні АП тому, що згідно з теоремою В. Глушкова про регуляризацію [18] довільний нерегулярний алгоритм можна перетворити в еквівалентний регулярний за допомогою послідовних, розгалужувальних та циклічних структур. Приклад такого перетворення при аналізі надійності діалогу «людина-комп'ютер» наведено в [26].

Успадковуючи ідеї публікацій [7, 79, 95-97] АП записуватимемо мовою алгоритмічних алгебр Глушкова [18]. Ця алгебра є четвіркою

$$\langle \mathbf{U}, \mathbf{B}, \Omega_1, \Omega_2 \rangle,$$

де  $\mathbf{U} = \{A, B, C, \dots\}$  – множина операторів;

$\mathbf{B} = \{\alpha, \beta, \gamma, \dots\}$  – множина логічних умов;

$\Omega_1$  – множина операцій, які породжують логічні умови з множини  $\mathbf{B}$ ;

$\Omega_2$  – множина операцій, які породжують оператори з множини  $\mathbf{U}$ .



Рис. 1.1. Класифікація АП

Оператор – це відображення інформаційної множини про стан системи в себе, тобто перетворення одного стану системи в інший.

Умова – це відображення поточного стану системи в двохелементну множину  $\{1, 0\}$ , де 1 – істина; 0 – хибність.

До операцій з множини  $\Omega_1$ , які породжують логічні умови, відносяться булеві операції диз'юнкції, кон'юнкції та заперечення.

До операцій з множини  $\Omega_2$ , які породжують оператори, відносяться:

- композиція  $A_1 A_2$  – бінарна операція, що породжує оператор, який полягає в послідовному застосуванні операторів  $A_1 \in \mathbf{U}$  і  $A_2 \in \mathbf{U}$  за порядком запису;
- $\omega$ -диз'юнкція  $(A_1 \vee A_2)$  – тернарна операція, що породжує оператор, який дорівнює  $A_1 \in \mathbf{U}$  при  $\omega=1$ , та дорівнює  $A_2 \in \mathbf{U}$  при  $\omega=0$ ,  $\omega \in \mathbf{B}$ ;
- $\omega$ -ітерації  $\{A\}$  – бінарна операція, що полягає в циклічному застосуванні оператора  $A$  (при хибному  $\omega$ ) до тих пір, поки умова  $\omega$  не стане істиною.

При моделюванні надійності АП використовують показники безпомилковості, тривалості та вартості. В залежності від типу АП ці показники інтерпретуються як:

- ймовірність досягнення мети, достовірність перетворення та передачі інформації, рівень бездефектності продукту праці, ймовірності відсутності помилок кожного типу, ймовірність своєчасного виконання алгоритму;
- трудомісткість технологічного процесу, тривалість виконання задачі, тривалість перетворення інформації;
- собівартість виробництва продукції, витрати на перетворення чи передачу інформації, витрати на виконання задачі, вартість досягнення мети.

Ресурси на виконання АП можуть бути обмеженими, наприклад, загальна тривалість чи вартість процесу, або кількість ітерацій циклічних структур – тобто число запасних заготовок в технологічному процесі виробництва продукції, кількість спроб введення пін-коду при отриманні готівки через банкомат або кількість Perezдач екзамену. При вичерпанні ресурсів АП переривається. В монографії розглядатимемо АП з нескінченними ресурсами.

Будемо розглядати АП з чітким характером помилок, який одночасно знаходиться лише в одному з двох станів – без помилок та з помилками. Тобто, вважатимемо, що наперед відомо, що є помилкою функціонування, і відповідно існує чітка границя між станами правильного та помилкового виконання. Процеси з нечітким характером помилок [84], тобто з розмитою границею між правильним та неправильним виконанням, не входять в об'єкт дослідження.

За стаціонарністю структури АП поділимо на процеси з фіксованою та гнучкою структурами. За фіксованої структури послідовність

виконання операцій або їх склад не залежать від тривалості або вартості деякого фрагмента АП, а за гнучкої структури – залежать. Прикладом АП з гнучкою структурою є виконання студентом курсової роботи. Студент при наближенні до встановлених термінів через дефіцит часу може змінити структуру АП, вилучивши з нього деякі операції, наприклад, самоконтроль курсової роботи. Формально гнучкої структури АП можна позбутися, якщо ввести в алгоритм функціонування додаткові логічні умови.

За стаціонарністю початкових даних АП поділимо на процеси з постійними та змінними характеристики. В останньому випадку характеристики надійності елементів процесу залежать від тривалості певного фрагмента АП або від кількості повторів деяких операторів чи логічних умов. Наприклад, тривалість та безпомилковість повторного введення пароля людиною-оператором може відрізнятися від характеристик першої спроби; або при дефіциті часу тривалість і якість виконання операцій зазвичай зменшуються. Нестаціонарності початкових даних можна позбутися перетворивши початковий АП за ідеями регуляризації алгоритмів. Для цього на кожне значення характеристики надійності створюємо окремий алгоритмічний елемент і за додатковими логічними умовами обираємо релевантні реалізації операцій. Надалі розглядатимемо АП з фіксованою структурою та з постійними характеристики надійності елементів.

Відповідно до концепції врахування помилок АП поділяються на одновимірні та багатовимірні. В одновимірних АП алгоритм може бути виконано з двома наслідками: 0 – з помилками або 1 – без помилок. В багатовимірному АП можливі  $m+1$  наслідки: 1 – без помилок,  $0^1$  – помилка першого типу,  $0^2$  – помилка другого типу, ...,  $0^m$  – помилка  $m$ -го типу (рис. 1.2). Таким чином, *багатовимірним* АП назвемо такий процес функціонування, при виконанні якого можуть з'являтися, виявлятися та виправлятися помилки різних типів.

Прикладами багатовимірних АП є технологічний процес виробництва і контролю якості РЕА [79], процес фільтрації води [167], процес функціонування системи «оператор–ЕОМ» [39, 102] та інші. Детальний опис цих АП наведено в табл. 1.1.

Для оцінювання надійності функціонування АП найчастіше використовують такі показники:

$p^1$  – ймовірність відсутності будь-якої помилки на виході АП;

$p^{0j}$  – ймовірність помилки  $j$ -го типу ( $j = \overline{1, m}$ );

$T$  – середній час виконання АП;

$C$  – середня вартість виконання АП.

Таблиця 1.1

## Приклади багатовимірних АП

Назва АП	Приклади основних операцій	Вхідні дані	Кінцевий результат	Типи помилок
Технологічний процес виробництва РЕА	<ol style="list-style-type: none"> <li>1. Підготовка купованих елементів</li> <li>2. Виготовлення комплектуючих.</li> <li>3. Набивка та пайка друкованих плат</li> <li>4. Складання та регулювання вузлів</li> </ol>	Сировина, матеріали, комплектуючі елементи	РЕА	<ol style="list-style-type: none"> <li>1. Порушення порядку розміщення радіоелементів</li> <li>2. Неправильне розміщення мікросхеми</li> <li>3. Розрив контактної доріжки</li> <li>4. Неякісна пайка</li> <li>5. Неякісна металізація отворів</li> </ol>
Процес фільтрації води	<ol style="list-style-type: none"> <li>1. Механічне очищення води</li> <li>2. Виведення змулених частинок і частинок, що осідають</li> <li>3. Біологічне очищення</li> <li>4. Бактеріологічне очищення</li> </ol>	Забруднена вода	Очищена вода	<ol style="list-style-type: none"> <li>1. Наявність піску та каменню</li> <li>2. Наявність змулених частинок</li> <li>3. Наявність азоту, фосфору та сірководню</li> <li>4. Наявність бактерій</li> </ol>
Система «оператор-ЕОМ» при продажі квитків	<ol style="list-style-type: none"> <li>1. Запит клієнта</li> <li>2. Набір запиту на клавіатурі</li> <li>3. Обробка запиту на ЕОМ</li> <li>4. Друкування квитка</li> </ol>	Запит клієнта	Квиток	<ol style="list-style-type: none"> <li>1. Помилка дати поїздки</li> <li>2. Помилки в паспортних даних клієнта</li> <li>3. Неправильний номер рейсу</li> </ol>

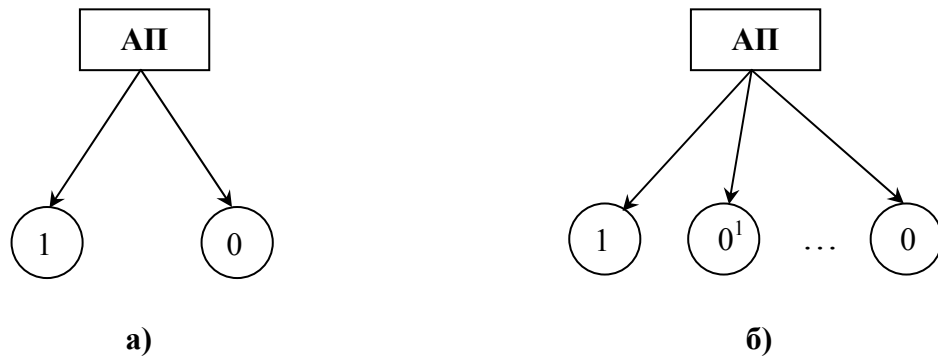


Рис. 1.2. Наслідки виконання:  
а) одновимірного АП; б) багатовимірного АП

При проектуванні АП за критерієм надійності виникає необхідність в розв'язанні:

– *задачі аналізу*, яка полягає в прогнозуванні показників надійності на основі структури АП та ймовірнісних, часових та вартісних характеристик операторів та логічних умов;

– *задачі синтезу*, яка полягає в генеруванні такого варіанта АП, який забезпечує необхідні рівні показників надійності.

На практиці найбільш поширеними задачами синтезу АП є [12, 38, 52, 75, 76, 93]:

- розподіл контрольних точок;
- вибір кратностей та часток контролю;
- вибір способів реалізації операцій процесу.

## 1.2. Методи оцінювання та забезпечення надійності алгоритмічних процесів

Перші роботи з оцінювання АП з'явилися закордоном на початку 1960-х років. Більшість моделей з ранніх досліджень спрямовані на розрахунок лише тривалості або вартості АП, та не враховують безпомилковості його виконання. Серед цих робіт найбільш відомими є мережі ПЕРТ С. Ейснера [130] та Х. Ельмаграбі [131], мережі ГЕРТ Притскера [101] та моделі комп'ютерних програм Б. Байцера [2].

В 70-х роках ХХ ст. в відкритій пресі з'являються перші роботи радянських вчених з моделювання надійності АП. Це роботи І. Сафонова [7, 94–97] з моделювання процесів функціонування ЕОМ і АСУ, А. Губінського [23, 24] з моделювання процесів функціонування людино-машинних систем та Б. Вігмана [10, 11] і Г. Дружиніна [29–31] з моделювання технологічних процесів. Спільною рисою цих робіт є

оцінювання показників надійності АП за допомогою типових структур з основних та допоміжних операцій. В них показниками надійності обрано ймовірність безпомилкового виконання АП, тривалість та вартість. Для моделювання надійності АП використовували такі типові структури: «послідовна», «паралельна», «робота–контроль–добробка», «робота–контроль–переробка», «робота–контроль–діагностика» та інші. В деяких роботах враховувалися як помилки 1-го, так і 2-го родів при контролі, тобто помилки типу «хибна тривога» та «пропуск цілі» [49]. Більшість алгоритмічних структур складаються з таких елементів [79]:

- *робочий оператор  $A$* , підчас виконання якого помилки можуть бути внесені, але не виявлені і не усунені;
- *добробка  $R$* , підчас виконання якої усуваються раніше внесені помилки, при цьому не виключається поява нових помилок;
- *добробка  $U$* , підчас виконання якої усуваються раніше внесені помилки, при цьому нові помилки не вносяться;
- *оновлення  $Z$* , підчас якого замінюється предмет діяльності на новий;
- *тотожний оператор  $E$* , який не змінює стан системи;
- *контроль  $\omega$* , підчас виконання якого внесені помилки виявляються.

На початку 1980-х А. Півоваров [62] запропонував моделі оцінювання достовірності та середньої тривалості обробки інформації в АСУ з використанням апарату теорії ймовірності та графів подій. Пізніше А. Маміконов [54] розробив моделі оцінки достовірності, середнього часу та вартості обробки інформації послідовних та циклічних алгоритмічних структур при незалежних, марківських та напівмарківських процесах виникнення помилок.

Закордонні дослідження в цій області зосереджені на аналізі надійності технологічних процесів [118, 119, 125, 128, 134, 136, 140, 143, 146, 148, 150, 157, 160, 161, 173, 176, 178, 181–183]. Причому суттєво переважають прикладні дослідження з прогнозування показників бездефектності, трудомісткості та прибутковості з урахуванням особливостей конкретного технологічного процесу. Більшість досліджень виконують вчені з Китаю, Тайваню та Індії. Моделі з цих робіт розрізняються за: 1) типом процесу (послідовні чи складальні); 2) за типами вхідних, проміжних та вихідних та буферів; 3) за поводженням з забракованими виробами (добробка, заміна чи утилізація); 4) за урахуванням помилок 1-го та 2-го родів при контролі; 5) за типом інспекції (перевірка однієї операції чи усіх попередніх операцій).

Огляд методів моделювання надійності АП зведено в табл. 1.2. В ній посилання йде за прізвищем першого співавтора або наукового



керівника, якщо про це відомо. Винятком є стійкі авторські колективи, які вказані повністю. Англійською наведені прізвища іноземців, роботи яких не перекладено на українську або російську мови.

Таблиця 1.2

Класифікація методів моделювання надійності АП

Ознака	Значення	Дослідники та публікації
1	2	3
Тип АП	Формальні алгоритми	С. Ейснер [130], Х. Ельмаграбі [131], Притскер [101], І. Сафонов [94, 95], І. Кузьмін [50], О. Ротштейн та С. Штовба [82, 85, 164, 172].
	Інформаційні технології	А. Ашероф [1], Б. Байцер [2], Н. Зінгер [42], А. Маміконов [54], А. Півоваров [62], І. Сафонов [7, 96, 97]
	Технологічні (трудові) процеси	Б. Вігман [10, 11], А. Гарсія-Діас [136], Г. Дружинін [29–31], О. Ротштейн [70, 79, 164], В. Yum [183], С. Tang [176], D. Ballou та Н. Pazer [118, 119], D. Yao та S. Zheng [125, 181], E. Jewkes [148], G. Rabinowitz [160], Н. Gurnani [143], J. Foster та J. Vilalobos [134, 177, 178], J. Hsu [146], К. Kang [150], N. Deliman [128], S. Gunter [140], Т. Raz [161], W. Shin [173], Y. Narahari [157]
	Людино-машинні процеси	М. Гриф та Є. Цой [20], А. Губінський [22–26, 43, 64, 67], В. Євграфов [14], О. Ротштейн [71, 77, 164], П. Чабаненко [104]
Чіткі показники надійності	Ймовірність безпомилковості	Усі з названих, окрім С. Ейснера, Притскера та Б. Байцера
	Тривалість або вартість	Усі з названих
	Ймовірність переривання	А. Губінський [25, 26, 43, 67], В. Євграфов [14], О. Ротштейн [79]
	Ймовірності помилок кожного типу	О. Ротштейн [71, 77, 79]

Продовження табл. 1.2

1	2	3
Нечіткі показники надійності	Ймовірність безпомилковості	О. Ротштейн та С. Штовба [82, 85, 164, 172]
	Тривалість або вартість	О. Ротштейн та С. Штовба [82, 85, 164, 172]
	Ймовірність переривання	–
	Ймовірності помилок кожного типу	–
Тип помилок	Бінарні	Усі з названих
	Багатоарні несумісні	О. Ротштейн [71, 77, 79]
	Багатоарні сумісні	–
Помилки при контролі	Відсутні	А. Вострецов [12], І. Сафонов [7, 96, 97], J. Hsu [146], S. Gunter [140], C. Tang [176], E. Jewkes [148], H. Gurnani [143], Y. Narahari [157], G. Rabinowitz [160], D. Yao та S. Zheng [125, 181]
	Тільки 2-го роду	А. Гарсія-Діас [136], Г. Дружинін [29–31], А. Маміконов [54], W. Shin [173], N. Deliman [128]
	1-го та 2-го родів	Б. Вігман [10, 11], М. Гриф та Є. Цой [20], А. Губінський [22-26, 43, 64, 67], В. Євграфов [14], І. Кузьмін [49], О. Ротштейн [71, 77, 79, 164], В. Yum [183], J. Foster та J. Vilalobos [134, 177, 178], K. Kang [150], T. Raz [161]
Невизначеність початкових даних	Математичне сподівання	Усі з названих
	Дисперсія	А. Губінський [22, 25, 43, 67], О. Ротштейн [79], В. Євграфов [14], D. Yao та S. Zheng [125, 181]
	Ймовірнісні розподіли	О. Ротштейн [79]
	Нечіткі розподіли	О. Ротштейн та С. Штовба [82, 85, 164, 172]

В теоретичному плані найбільш результативними виявились дослідження наукової школи А. Губінського, які систематизовані в книгах [1, 14, 20, 22, 23, 43, 64, 79, 82]. Серед цих робіт виділимо монографії [79, 82], в яких запропоновано принципово нові підходи до моделювання надійності АП.

В книзі [79] та в роботах [71, 77], що їй передували, О. Ротштейн для врахування помилок різних типів запропонував матричні моделі надійності операторів, логічних умов та алгоритмічних структур. Таке багатоарне моделювання надійності потрібно для врахування факту про те, що ймовірності внесення, виявлення та усунення помилок різних типів є неоднаковими, так само як і витрати ресурсів на ці події. Матричне представлення характеристик безпомилковості дозволило отримати компактні моделі надійності алгоритмічних структур, які виявилися навіть простішими ніж з бінарними помилками.

В монографії О. Ротштейна і С. Штовби [82] та в роботах [85, 164, 172], що їй передували, запропоновано задавати нечіткими числами невизначені характеристики елементів АП, та відповідно розроблено нечіткі моделі надійності алгоритмічних структур. Нечіткі моделі отримані з чітких за принципом нечіткого узагальнення [58, 186].

Вважається, що нечіткі числа задаються в  $\alpha$ -формі

$$\tilde{y} = \bigcup_{\alpha \in [0,1]} (\underline{y}_\alpha, \bar{y}_\alpha), \quad (1.1)$$

де  $\underline{y}_\alpha$  та  $\bar{y}_\alpha$  – найменше та найбільше значення  $\alpha$ -зрізу нечіткого числа  $\tilde{y}$ .

За класифікацією з монографії [124] метод з [82], відноситься до класу *probist fuzzy reliability* – нечітка теорія надійності (*reliability*) з чіткими бінарними станами (*binary states*) системи та нечіткими ймовірностями (*fuzzy probability*) переходів між ними. За методом [82] можна прогнозувати нечіткі ймовірність правильного виконання та тривалість АП за експертних оцінок характеристик надійності типу «Низька безпомилковість», «Середня тривалість» тощо. Пізніше нечіткі моделі з [82] застосовувалися при прогнозуванні надійності АП в роботах [17, 19, 111].

Сучасний стан моделювання надійності алгоритмічних структур описано табл. 1.3.

В табл. 1.3 використовується алгоритмічна мова алгебри В. Глушкова для запису послідовних, паралельних, розгалужувальних та циклічних алгоритмічних структур.

Таблиця 1.3

Моделі надійності алгоритмічних структур  
(*Crisp* – чіткі моделі; *Fuzzy* – нечіткі моделі; «+» – є; «-» – немає)

Алгоритмічні структури		Бінарні помилки		Багатоарні помилки	
		<i>Crisp</i>	<i>Fuzzy</i>	<i>Crisp</i>	<i>Fuzzy</i>
Послідовні	$A_1 A_2$	+	+	+	-
	$A^N$	+	+	+	-
Паралельні	$[A_1, A_2]$	+	-	+	-
	$\langle A_1, A_2 \rangle$	+	-	-	-
Розгалужувальні	$(A_1 \vee A_2)$ $\omega$	+	+	+	-
	$A(E \vee U)$ $\omega$	+	-	+	-
	$A(E \vee R)$ $\omega$	+	+	+	-
	$A(E \vee (E \vee U))$ $\psi \quad \omega$	+	-	+	-
	$A(E \vee (E \vee R))$ $\psi \quad \omega$	+	+	+	-
Циклічні	$\{ZA\}$ $\omega$	+	+	+	-
	$A\{E \vee U\}$ $\omega$	+	-	+	-
	$A\{E \vee R\}$ $\omega$	+	-	+	-

Нижче наведено визначення алгоритмічних структур з табл. 1.3:

- $A_1 A_2$  – виконання робочих операторів  $A_1$  та  $A_2$  в порядку запису;
- $A^N$  – виконання робочого оператора  $A$  рівно  $N$  раз;
- $[A_1, A_2]$  – паралельне виконання різних задач операторами  $A_1$  та  $A_2$ ;

- $\langle A_1, A_2 \rangle$  – паралельне виконання однієї задачі операторами  $A_1$  та  $A_2$ ;
- $(A_1 \vee A_2)$  – виконання робочого оператора  $A_1$  за істинності умови ( $\omega=1$ ) та робочого оператора  $A_2$  в протилежному випадку, тобто коли  $\omega=0$ ;
- $A(E \vee U)$  та  $A(E \vee R)$  – виконання робочого оператора  $A$ , контролю  $\omega$  та доробки ( $U$  або  $R$ ) при виявленні помилок;
- $A(E \vee (E \vee U))$  та  $A(E \vee (E \vee R))$  – виконання робочого оператора  $A$ , який з ймовірністю  $p_\psi$  потрапляє на контроль  $\omega$  та при виявленні помилок на доробку ( $U$  або  $R$ );
- $\{ZA\}$  – циклічне виконання оператора оновлення  $Z$  та робочого оператора  $A$ . Цикл закінчується якщо контролем  $\omega$  не знайдено помилок виконання останнього оператора  $A$ ;
- $A\{E \vee U\}$  та  $A\{E \vee R\}$  – виконання робочого оператора  $A$  та циклу контроль  $\omega$  – доробка ( $U$  або  $R$ ) при виявленні помилок.

З табл. 1.3 зрозуміло, що на сьогодні відсутні нечіткі моделі надійності алгоритмічних структур для процесів з багатоарними помилками.

Оптимізувати надійність АП можна шляхом варіювання: а) переліком та кількістю контрольних процедур; б) варіантами реалізацій операторів та логічних умов; в) факторами, що впливають на характеристики надійності операторів та логічних умов. Шляхи (а) та (б) назвемо *структурною оптимізацією* АП, а шлях (в) – *параметричною оптимізацією*.

Серед задач структурної оптимізації найбільш популярними є оптимізація контролей – розстановка контрольних точок та вибір кратності контролю. Задачі оптимізації контролю полягають в забезпеченні екстремальних значень одного показника надійності при обмеженнях на інший. Їх аналогами в класичній дискретній оптимізації є задача про рюкзак [154]. Відповідність між задачами є такою: предмети рюкзак – операції контролю; об'єм рюкзак – вартість або тривалість АП; корисність рюкзак – безпомилковість АП. Відмінність полягає в характері цільової функції та обмежень; вони лінійні в задачі про рюкзак та нелінійні в задачі оптимізації надійності АП. Таким чином задача розстановки контрольних точок є нелінійним аналогом задачі про рюкзак. Задача про рюкзак є NP-складною [137], тому і складність задачі

розстановки контрольних точок є щонайменше такою ж.

Задачу структурної оптимізації АП назвемо *загальною*, якщо в ній поєднано розстановку контрольних точок (чи встановлення кратності контролю) з вибором реалізацій операторів та логічних умов. *Структурно-параметричною* назвемо оптимізацію АП, яка поєднує керовані змінні з задач структурної та параметричної оптимізації. Класифікація методів оптимізації надійності АП зведена в табл. 1.4. В ній посилання на авторів йде за правилами формування табл. 1.2.

Таблиця 1.4

Класифікація методів оптимізації надійності АП

Ознака	Значення	Дослідники та публікації
1	2	3
Задача оптимізації	Розташування контролів	Б. Вігман [10, 11], А. Вострецов [12], М. Гриф та Є. Цой [20], А. Губінський [43], В. Євграфов [14], Н. Зінгер [42], С. Заносін [38], В. Лопухін [52], О. Ротштейн [75, 76, 79]
	Вибір реалізацій операцій	Б. Вігман [10, 11], М. Гриф та Є. Цой [19–21], А. Губінський [24, 43], В. Євграфов [14], І. Кузьмін [50], І. Рібаков [93], І. Сафонов [7, 96, 97], О. Ротштейн [70, 79, 82, 172]
	Загальна задача структурної оптимізації	Б. Вігман [10, 11]
	Параметрична оптимізація	–
	Структурно-параметрична оптимізація	–
Метод оптимізації	Повний перебір	Б. Вігман [10, 11], Н. Зінгер [42]
	Динамічне програмування	М. Гриф та Є. Цой [20], А. Гарсія-Діас [136], А. Губінський [43], В. Євграфов [14], І. Рібаков [93], І. Сафонов [7, 96, 97], С. Tang [176], D. Yao та S. Zheng [181], Н. Gurnani [143], J. Foster та J. Vilalobos [178], J. Hsu [146], S. Gunter [140]

Продовження табл. 1.4

1	2	3
Метод оптимізації	Гілок та меж	О. Ротштейн [70, 79, 82], Т. Raz [161], В. Yum [183]
	Лінійне програмування	М. Гриф та Є. Цой [19-21], А. Губінський [43], В. Євграфов [14], А. Маміконов [54]
	Евристики	М. Гриф та Є. Цой [19-21], А. Губінський [43], В. Лопухін [52], О. Ротштейн [75, 76, 79], J. Foster та J. Vilalobos [134], К. Kang [150]
Критерії оптимальності	Безпомилковість	А. Губінський [24, 43], В. Євграфов [14], Н. Зінгер [42], А. Маміконов [54], О. Ротштейн [70, 75, 76, 79, 82, 172], І. Сафонов [7, 96, 97]
	Тривалість	А. Губінський [43], В. Євграфов [14], А. Маміконов [54], О. Ротштейн [70, 75, 76, 79, 82, 172], І. Сафонов [7, 96, 97], Н. Gurnani [143], Y. Narahari [157], G. Rabinowitz [160]
	Вартість або прибуток	А. Вострецов [12], А. Гарсія-Діас [136], А. Губінський [24, 43], С. Заносін [38], В. Євграфов [14], В. Лопухін [52], А. Маміконов [54], І. Рібаков [93], О. Ротштейн [70, 75, 76, 79, 82, 172], В. Yum [183], С. Tang [176], D. Ballou та Н. Pazer [118, 119], D. Yao та S. Zheng [125, 181], E. Jewkes [148], G. Rabinowitz [160], Н. Gurnani [143], J. Foster та J. Vilalobos [134, 177, 178], J. Hsu [146], К. Kang [150], N. Deliman [128], S. Gunter [140], Т. Raz [161], W. Shin [173], Y. Narahari [157]
	Ймовірність помилок кожного типу	–
	Ймовірність переривання	–
	Кількість контролю	–

Продовження табл. 1.4

1	2	3
Обмеження	Безпомилковість	А. Губінський [43], В. Євграфов [14], А. Маміконов [54], І. Рібаков [93], О. Ротштейн [70, 75, 76, 79, 82, 172], І. Сафонов [7, 96, 97], J. Foster та J. Vilalobos [134]
	Тривалість	А. Губінський [43], В. Євграфов [14], А. Маміконов [54], О. Ротштейн [70, 75, 76, 79, 82, 172], І. Сафонов [7, 96, 97], J. Foster та J. Vilalobos [134, 178], В. Yum [183], W. Shin [173], Н. Gurnani [143], D. Yao та S. Zheng [181],
	Вартість або прибуток	А. Губінський [43], В. Євграфов [14], А. Маміконов [54], О. Ротштейн [70, 75, 76, 79, 82, 172]
	Ймовірність помилок кожного типу	–
	Ймовірність переривання	–
	Кількість контролю	–
Мета	Чітка	Усі з названих
	Нечітка	–
Обмеження	Чіткі	Усі з названих
	Нечіткі	О. Ротштейн та С. Штовба [82, 172]
Цільова функція	Чітка	Усі з названих
	Нечітка	О. Ротштейн та С. Штовба [82, 172]

З табл. 1.4 видно, що дослідження зосереджені на вирішенні часткових задач структурної оптимізації АП за критеріями безпомилковості, тривалості та вартості (або прибутковості). При оптимізації не враховуються такі важливі для деяких АП показники як ймовірності відсутності помилок кожного типу, ймовірність переривання АП та кількість контролерів. Останній показник особливо важливий при розробці АП з системою автоматизованого контролю, коли кожен контрольний пост реалізується апаратно. Зовсім не досліджені задачі



параметричної оптимізації АП та змішані задачі структурно-параметричної оптимізації. Такий стан пояснюється складністю задач оптимізації. Навіть найлегша з них – розстановка контрольних точок є NP-складною. Тому розробка алгоритмів вирішення цих задач на базі класичних методів оптимізації є вкрай складною.

Вперше задачі нечіткої оптимізації надійності АП запропоновані в [82]. Вони стосуються одного класу задач структурної оптимізації АП, а саме – вибору варіантів реалізації операторів та логічних умов за критеріями нечіткої безпомилковості та нечіткої тривалості. Оптимізація здійснюється за узагальненим на випадок нечітких початкових даних алгоритмом гілок та меж з [79]. Алгоритм оптимізації формує паретівську множину шляхом відсікання неперспективних фрагментів. Для цього застосовуються тривіальні правила порівняння альтернатив з нечіткими показниками надійності. Вибір розв'язку з паретівської множини альтернатив здійснюється після дефазифікації нечітких показників надійності. Інші задачі нечіткої оптимізації надійності АП досі не сформульовані.

Зважаючи на те, що нечітка оптимізація АП майже не досліджена, проведемо огляд можливих варіантів її розв'язання. В теорії нечітких множин задача оптимізації вважається нечіткою якщо виконується хоча б одна із таких умов:

- значенням цільової функції або обмежень є нечіткі числа;
- праві частини обмежень задані нечіткими числами;
- мета оптимізації задана нечіткими числами.

Математичні задачі нечіткої оптимізації досліджували багато науковців, зокрема С. Орловський [59], В. Тарасов [58 глава 9], О. Воцінін [13], Ю. Зайченко [36, 37], М. Згуровський [41], А. Язенін [112, 113], Б. Лю [53], Н. Zimmerman [186]. В більшості робіт вважається, що функція належності нечітких чисел є трикутною (рис. 1.3).

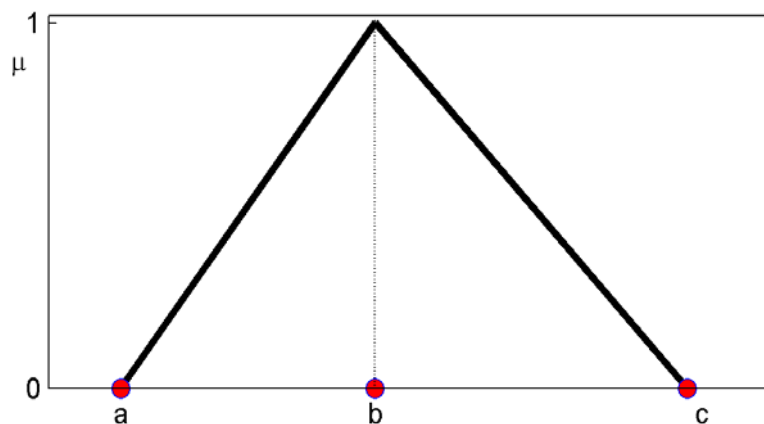


Рис. 1.3. Нечітке число з трикутною функцією належності ( $\mu$  – ступінь належності)

При нечіткій оптимізації потрібно вміти відповідати на питання:

- 1) чи задовольняє обмеження поточний вектор керованих змінних;
- 2) чи досягає екстремуму цільова функція для поточного вектора керованих змінних;
- 3) як діяти, щоб з поточної точки потрапити в оптимум.

Аналогічні питання виникають і при чіткій оптимізації [69], але відповідь на перше з них є тривіальним. В нечіткому випадку кожне питання вирішується неоднозначно через необхідність порівняння нечітких чисел. Відношення порядку над нечіткими числами  $\tilde{A}$  та  $\tilde{B}$  однозначно можна встановити лише в тривіальному випадку, коли

$$\forall \alpha \in [0, 1]: (\underline{A}_\alpha \leq \underline{B}_\alpha \text{ та } \bar{A}_\alpha \leq \bar{B}_\alpha), \quad (1.2)$$

де  $[\underline{A}_\alpha, \bar{A}_\alpha]$  та  $[\underline{B}_\alpha, \bar{B}_\alpha]$  –  $\alpha$ -зрізи цих нечітких чисел. Експериментальні дослідження, які проведені [82] показали, що при оптимізації реальних АП за правилом (1.2) відсікається небагато варіантів, тому на виході отримуємо велику паретівську множину альтернатив. Звідси ранжувати варіанти АП доводиться за евристичними правилами порівняння нечітких чисел. Аналогічна проблема виникає при порівнянні чіткого числа з нечітким.

В нечіткій оптимізації склалися 2 підходи. За першим підходом намагаються перетворити нечітку оптимізацію в чітку [13, 36, 53, 58, 59, 112, 113, 115, 186]. Для цього вводять евристичні правила порівнянь нечітких чисел, або оптимізують на деякому  $\alpha$ -рівні функції належності. За другим підходом [37, 58, 186] застосовують принцип Беллмана–Заде про злиття цілей та обмежень з нечіткої теорії прийняття рішень [4, 121].

*Перший підхід* схожий на класичні пошукові методи оптимізації – чіткі постановки розмивають за принципом нечіткого узагальнення. Таке розмиття задачі оптимізації має за мету пом'якшити обмеження, сформувавши нечітку область допустимих розв'язків. Спочатку розберемось, якими чином треба ввести нечіткі числа в обмеження задачі оптимізації, щоб сформувати коректну нечітку область допустимих розв'язків. Для прикладу розглянемо чітке обмеження з вартості АП

$$C(\mathbf{X}) \leq 9, \quad (1.3)$$

яке, спробуємо пом'якшити механічною заміною чіткого числа в правій частині на нечітке число. В результаті отримуємо такий вираз:

$$C(\mathbf{X}) \leq \tilde{9}, \quad (1.4)$$

де  $\tilde{9}$  – нечітке число з відомою на момент оптимізації функцією на-

лежності.

Припустимо, що нечітке число  $\tilde{9}$  задано трикутною функцією належності (рис. 1.3) з параметрами  $a=7, b=9, c=11$ . Тоді, за визначенням операцій порядку над нечіткими числами [4], умова (1.4) буде виконана, якщо  $C(\mathbf{X}) \in [0, 7]$  (рис. 1.4а), що з урахуванням невід'ємності цього показника можна записати як  $C(\mathbf{X}) \leq 7$ . Таким чином, умова (1.4) з нечіткою правою частиною не тільки не пом'якшила обмеження (1.3), а зробило його більш жорстким.

Для пом'якшення умови (1.3) праву частину задають так званим нечітким числом з насиченням, ядро якого складають усі числа, що не перевищують 9. Функція належності нечіткого числа з насиченням є монотонною. В аналізованому випадку ядро нечіткого числа з насиченням дорівнюватиме  $[0, 9]$  (рис. 1.4б). При  $C(\mathbf{X}) \in [0, 9]$  ступінь виконання такої нечіткої умови дорівнюватиме 1, тобто нечітке обмеження задовольняється повною мірою. Значення  $C(\mathbf{X}) \in [0, 9]$  задовольняють і аналогічне чітке обмеженню (1.2). При більших значеннях  $C(\mathbf{X})$  ступінь виконання нечіткої умови визначається функцією належності нечіткого обмеження.

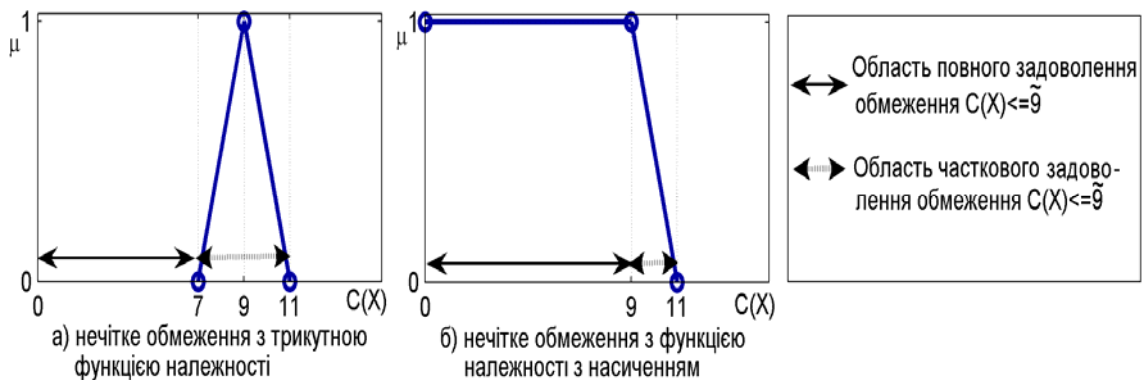


Рис. 1.4. Нечіткі обмеження  $C(\mathbf{X}) \leq \tilde{9}$

Узагальнюючи на довільні значення правих частин, сформулюємо такі правила перетворення чітких обмежень-нерівностей в нечіткі умови

$$\left. \begin{aligned} a \geq A &\rightarrow a = \tilde{A}, \text{ де } core(\tilde{A}) = [A, \bar{A}] \\ a > A &\rightarrow a = \tilde{A}, \text{ де } core(\tilde{A}) = [A + \varepsilon, \bar{A}] \\ a \leq A &\rightarrow a = \tilde{A}, \text{ де } core(\tilde{A}) = [\underline{A}, A] \\ a < A &\rightarrow a = \tilde{A}, \text{ де } core(\tilde{A}) = [\underline{A}, A - \varepsilon] \end{aligned} \right\}, \quad (1.5)$$

де  $\underline{A}$  та  $\bar{A}$  – найменш та найбільш можливі значення змінної  $a$ ;

$\varepsilon$  – мала величина;

$core(\tilde{A})$  – ядро нечіткого числа  $\tilde{A}$ .

Припустимо, що в постановці задачі оптимізації є  $V$  нечітких умов. Позначимо ступені виконання цих умов розв'язком  $\mathbf{X}$  через  $\mu_1(\mathbf{X}), \mu_2(\mathbf{X}), \dots, \mu_V(\mathbf{X})$ . Виникає питання як врахувати таку інформацію при прийнятті рішення. На нашу думку тут можливі такі два способи.

*Перший спосіб* стосується встановлення відношення порядку над розв'язками задачі оптимізації. Для цього використовують такі тривіальні правила:

- якщо значення цільової функція для  $\mathbf{X}_1$  краще ніж для  $\mathbf{X}_2$  та  $\mu_v(\mathbf{X}_1) \geq \mu_v(\mathbf{X}_2)$ ,  $v = \overline{1, V}$ , тоді  $\mathbf{X}_1$  краще за  $\mathbf{X}_2$ ;
- якщо значення цільової функція для  $\mathbf{X}_1$  не гірше ніж для  $\mathbf{X}_2$ , та  $\forall v: \mu_v(\mathbf{X}_1) \geq \mu_v(\mathbf{X}_2)$ , та  $\exists v: \mu_v(\mathbf{X}_1) > \mu_v(\mathbf{X}_2)$ ,  $v \in \{1, 2, \dots, V\}$ , тоді  $\mathbf{X}_1$  краще за  $\mathbf{X}_2$ .

Для нетривіальних випадків відношення порядку між розв'язками без залучення евристик встановити неможливо, тому використання такого способу є обмеженим.

*Другий спосіб* стосується визначення області допустимих розв'язків. Пропонується визначити  $V$ -місну функцію, яка б відображала  $(\mu_1(\mathbf{X}), \mu_2(\mathbf{X}), \dots, \mu_V(\mathbf{X}))$  в двохелементну множину  $\{0 - \text{недопустимий розв'язок}, 1 - \text{допустимий розв'язок}\}$ .

Відображення  $(\mu_1(\mathbf{X}), \mu_2(\mathbf{X}), \dots, \mu_V(\mathbf{X})) \rightarrow \{0, 1\}$  можна реалізувати різними функціями згортки. Найпростіше це зробити, встановивши для кожного обмеження мінімальне порогове значення ступеня належності, досягнення якого розв'язком задачі оптимізації інтерпретується як виконання нечіткої умови. Позначимо через  $\mu_v^*$  – порогове значення ступеня належності  $v$ -ої нечіткої умови,  $v = \overline{1, V}$ . Тоді розв'язок буде допустимим, якщо

$$\forall v \in \{1, 2, \dots, V\}: \mu_v(\mathbf{X}) \geq \mu_v^*. \quad (1.6)$$

Функції належності нечітких умов та порогові значення ступеня належності відомі до початку оптимізації, тому їм можна поставити у відповідність звичайні чіткі обмеження. Для усунення цього недоліку можна крім порогових значень  $\mu_v^*$  за кожним нечітким обмеженням встановити і інтегральну планку  $\mu^*$

$$\begin{cases} \forall v \in \{1, 2, \dots, V\}: \mu_v(\mathbf{X}) \geq \mu_v^* \\ \sum_{v=1, V} \mu_v(\mathbf{X}) \geq \mu^* \end{cases}, \quad (1.7)$$

де  $\mu^* \in \left( \sum_{v=1, V} \mu_v^*, V \right)$ .

Але і в цьому випадку з (1.7) та з аналітичних виразів функцій належності можна записати еквівалентні обмеження без нечітких чисел. Відповідно, другий спосіб використання інформації про нечітке задоволення пом'якшених обмежень фактично не відрізняється від чіткого послаблення обмежень.

Таким чином, спроба сформулювати нечітку оптимізацію за принципом нечіткого узагальнення чітких постановок задач пошуку екстремуму призводить до залучення евристичних правил порівняння нечітких чисел, що вносить суб'єктивізм в ранжування альтернатив. Намагання обійтися без евристичних правил порівняння нечітких чисел вироджує задачу нечіткої оптимізації в задачу чіткого математичного програмування, тобто є завуальованою під нечіткі числа реалізацією чітких обмежень.

За *другим підходом* до нечіткої оптимізації при ранжуванні альтернатив застосовують принцип Беллмана–Заде. Замість знаходження екстремуму нечіткої цільової функції на нечіткій області допустимих розв'язків задачу оптимізації ставлять як пошук вектора керованих змінних  $\mathbf{X}$ , який більшою мірою задовольняє нечітку мету  $\tilde{G}$ . Нечітка мета може складатися з декількох нечітких критеріїв  $\tilde{C}_1, \tilde{C}_2, \dots, \tilde{C}_V$ .

Позначимо через  $\mu_{\tilde{C}_i}(\mathbf{X})$  – ступінь задоволення альтернативою  $\mathbf{X}$  нечіткого критерію  $\tilde{C}_i$ ,  $i = \overline{1, V}$ . Тоді за принципом Беллмана–Заде [4, 121] ступінь задоволення мети оптимізації за всіма критеріями одночасно розраховують таким чином:

$$\mu_{\tilde{G}}(\mathbf{X}) = \mu_{\tilde{C}_1}(\mathbf{X}) \wedge \mu_{\tilde{C}_2}(\mathbf{X}) \wedge \dots \wedge \mu_{\tilde{C}_V}(\mathbf{X}), \quad (1.8)$$

де  $\wedge$  –  $t$ -норма, яку зазвичай реалізують операцією мінімуму.

Як оптимальний розв'язок обирають альтернативу з максимальним ступенем належності (1.8).

Переваги нечіткої оптимізації за принципом Беллмана–Заде над оптимізацією за нечітким узагальненням полягають в: 1) простішій схемі прийняття рішення; 2) меншій обчислювальній складності; 3) меншому суб'єктивізмі, тому що є лише один евристичний пара-

метр – реалізація t-норми в (1.8). Як недолік, вкажемо на те, що досі залишаються відкритими такі питання:

- як розрахувати ступінь задоволення нечіткого критерію довільною альтернативою;
- за якими правилами генерувати альтернативи, щоб знайти оптимум за прийнятний час.

### **1.3. Принципи моделювання надійності алгоритмічних процесів**

Ґрунтуючись на основних ідеях моделювання надійності АП [2, 10, 11, 29-31] та методів оптимізації надійності АП [43] сформулюємо ряд методологічних принципів, які будемо використовувати для оцінювання та забезпечення надійності багатовимірних АП:

1. *Принцип опису АП мовою алгоритмічних алгебр.* У відповідності до цього принципу структуру довільного АП можна записати за допомогою системи алгоритмічних алгебр Глушкова [18].

2. *Принцип переходу від алгоритмічного до ймовірнісного опису.* Згідно з цим принципом, на основі алгоритмічного опису будуються ймовірнісні моделі, які дозволяють оцінити показники надійності АП за відомими характеристиками надійності операторів і логічних умов. Алгоритмічний опис відображає логіко-часову послідовність дій при функціонуванні системи. Елементами алгоритмічного опису є оператори і логічні умови. Ймовірнісним описом називаються отриманні на основі алгоритмічного опису формули розрахунку показників надійності за відомими характеристиками часу, витрат і ймовірності правильного виконання операторів та логічних умов.

3. *Принцип типових алгоритмічних структур.* Відповідно до цього принципу оцінка показників надійності АП здійснюється за такою методикою:

- виділення комбінацій операторів і логічних умов, які часто зустрічаються, в типові структури;
- розробка моделей надійності таких типових структур;
- оцінка показників надійності типових структур, шляхом заміни на єдиний оператор з еквівалентними характеристиками;
- послідовне застосування типових структур для редукції початкового АП до єдиного оператора.

4. *Принцип багатоваріантності АП.* Відповідно до цього принципу АП може перебувати в багатьох станах, які визначаються наявністю помилок різних типів. Після виконання будь-якого оператора чи логічної умови стани АП можуть бути такими: 1) відсутність будь-яких помилок функціонування; 2) наявність помилки функціонування

1-го типу; 3) наявність помилки функціонування 2-го типу тощо. За сумісних помилок АП може знаходитися також в станах з парними, трійними, четверними помилками тощо. Якщо помилки не розрізняються за типами, тоді АП може знаходитися в одному з двох станів – без помилок функціонування та з помилками.

5. *Принцип представлення експертних оцінок початкових даних нечіткими числами.* Відповідно до цього принципу ймовірнісні та часові характеристики надійності операторів та логічних умов задаються нечіткими числами у вигляді такої трійки [163, 164]

$$q = (\underline{q}, \bar{q}, l), \quad (1.9)$$

де  $\underline{q}$  ( $\bar{q}$ ) – нижня (верхня) межа зміни  $q$ ;

$l$  – експертна оцінка параметра  $q$  в діапазоні  $[\underline{q}, \bar{q}]$ , яка обирається з терм-множини {Низький, Середній, Вище Середнього, Високий}.

6. *Принцип  $\alpha$ -рівневого представлення нечітких чисел.* Відповідно до цього принципу для зручності розрахунків та компактності моделей нечіткі числа задаватиме  $\alpha$ -формі (1.2)

7. *Принцип нечіткого узагальнення моделей надійності.* Відповідно до цього принципу відомі моделі надійності типових алгоритмічних структур узагальнюються на випадок нечітких початкових даних, що дозволяє використовувати експертні оцінки.

8. *Принцип багатоваріантності постановок задач оптимізації надійності АП.* Відповідно до цього принципу задачі оптимізації надійності АП можна ставити в різний спосіб. Багатоваріантність постановок задач оптимізації полягає як в різних способах формування критеріїв і обмежень, так і різних методах покращення надійності АП. При цьому задачі оптимізації за характером керованих змінних можуть бути дискретними, неперервними та змішаними.

9. *Принцип генетичної оптимізації надійності АП.* Відповідно до цього принципу різноманітні задачі оптимізації надійності АП розв'язуватимемо з єдиних позицій за допомогою ГА. Особливістю ГА є можливість отримання гарних розв'язків для дискретних, неперервних та змішаних задач оптимізації без суттєвих модифікацій основних генетичних процедур.

10. *Принцип поєднання ГА з схемою Беллмана–Заде для нечіткої оптимізації надійності АП.* Відповідно до цього принципу в нечіткій оптимізації альтернативи ранжуватимемо за схемою Беллмана–Заде з нечіткої теорії прийняття рішень. Генерувати альтернативи будемо за допомогою ГА, який за рахунок імітації механізмів природної еволюції суттєво скорочує перебір варіантів.

## 2. МОДЕЛЮВАННЯ НАДІЙНОСТІ БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

В цьому розділі наводяться матричні моделі надійності операторів, логічних умов, типових алгоритмічних структур та алгоритм аналізу надійності багатовимірних алгоритмічних процесів. Матеріали розділу базуються на роботах [71, 77, 79].

### 2.1. Аналіз помилок різних типів

Помилки різних типів, що виникають в АП, пов'язані з неправильним виконанням людиною (оператором, технологом, контролером) або технікою деякого фрагмента алгоритму, тобто послідовності операцій, дій або робіт. В більшості робіт з теорії надійності алгоритмів [30, 96] врахування помилок здійснюється за бінарною концепцією, де розрізняються лише два стани виконання алгоритму з помилками або без помилок. Самі ж помилки не розрізняються за типом, тобто не важливо, яка помилка зроблена при виконанні алгоритму. В багатьох реальних задачах використання бінарної концепції врахування помилок є недоцільним, оскільки для різних типів помилок різними є ймовірності їх внесення, виявлення та усунення, так само як і витрати ресурсів на ці процедури. Наприклад, для процесу виробництва друкованих плат ймовірність виправлення дефектів травлення, пов'язаних з неправильним розташуванням фотошаблону, майже завжди дорівнює нулю, а ймовірність виправлення дефекту, пов'язаного з неповною очисткою плати від відпрацьованого фоторезисту є близькою до одиниці. Врахування помилок різних типів дозволяє ефективніше проектувати АП. Проілюструємо це на прикладі технологічного процесу виробництва РЕА.

*Приклад 2.1* Розглядається такий фрагмент процесу виготовлення РЕА

$$B = A \underset{\omega}{(E \vee U)}, \quad (2.1)$$

де  $A$  – пайка друкованої плати;

$\omega \in \{\omega_1, \omega_2\}$  – множина контролів якості виконання операції  $A$ ;

$U$  – доробка, яка виправляє дефекти, що виявлені контролем  $\omega$ .

Будемо вважати, що під час пайки можуть бути внесені 2 типа дефектів: 1 – непропай; 2 – перегрів мікросхеми. Ймовірності внесення 1-го та 2-го типу дефектів такі:  $p_A^{01}=0,15$ ;  $p_A^{02}=0,05$ . Вартість вико-



нання пайки  $c_A=20$ .

Будемо вважати, що при контролі відсутні помилки першого роду («хибна тривога»). Ймовірно-вартісні характеристики контролів  $\omega$  наведені в табл. 2.1. В цій таблиці через  $k_{\omega}^{001}$  та  $k_{\omega}^{002}$  позначено ймовірності виявлення дефектів 1-го та 2-го типу, а через  $c_{\omega}$  – вартості виконання контролів  $\omega$ .

Таблиця 2.1

Ймовірно-вартісні характеристики контролів  $\omega$

Варіант	$k_{\omega}^{001}$	$k_{\omega}^{002}$	$c_{\omega}$
$\omega_1$	0,88	0,40	18
$\omega_2$	0,75	0,80	20

При виконанні доробки дефекти виправляються з ймовірністю  $\nu_U^1=0,98$ . Середня вартість виконання доробки  $c_U=80$ .

Ціна експлуатації РЕА з дефектами 1-го та 2-го типу така:  $c_1=10$ ;  $c_2=300$ .

Необхідно визначити, який контроль краще встановити  $\omega_1$  чи  $\omega_2$  та порівняти прийняття рішення за бінарною та  $n$ -арною концепцією врахування помилок.

*Розв'язок.* Визначимо ймовірно-вартісні характеристики алгоритму (2.1). Для цього розрахуємо ймовірності появи дефектів 1-го та 2-го типу на виході алгоритму (2.1) за такими формулами

$$p_B^{01} = p_A^{01} \cdot (1 - k_{\omega}^{001} \cdot \nu_U^1),$$

$$p_B^{02} = p_A^{02} \cdot (1 - k_{\omega}^{002} \cdot \nu_U^1).$$

Середня вартість виконання алгоритму (2.1) розраховується так

$$c_B = c_A + c_{\omega} + c_U \cdot (p_A^{01} \cdot k_{\omega}^{001} + p_A^{02} \cdot k_{\omega}^{002}).$$

В табл. 2.2 наведено ймовірно-вартісні характеристики різних варіантів виконання алгоритму (2.1). В цій таблиці через  $p_B^1$  позначено ймовірність безпомилкового виконання (2.1), яку розраховано так  $p_B^1 = 1 - p_B^{01} - p_B^{02}$ .

Розглянемо прийняття рішення за бінарною концепцією врахування помилок. Безпомилковість варіантів виконання алгоритму (2.1)

є однаковою  $p_B^1=0,95$ , але середня вартість виготовлення РЕА з контролем  $\omega_1$  є дешевшою і тому цей варіант є оптимальним.

Таблиця 2.2

Ймовірнісно-вартісні характеристики алгоритму (2.1)

Варіант	$p_B^1$	$p_B^{01}$	$p_B^{02}$	$c_B$
$B = A ( E \vee U )$ $\omega_1$	0,95	0,02	0,03	50,2
$B = A ( E \vee U )$ $\omega_2$	0,95	0,04	0,01	52,2

При багатоарній концепції враховують не тільки бездефектність і вартість процесу, але і ймовірності дефектів різних типів. З табл. 2.2 видно, що хоча бездефектність є однаковою для обох варіантів, але в першому варіанті «дорогих» дефектів (другий тип) в тричі більше ніж в другому варіанті. Тому як критерій прийняття рішення будемо використовувати сумарні витрати на виробництво та експлуатацію РЕА. Вони розраховуються таким чином

$$c = c_B + (p_B^{01} \cdot c_1 + p_B^{02} \cdot c_2).$$

Загальні витрати для першого та другого варіантів склали 59,4 та 55,6. Таким чином варіант виконання алгоритму (2.1) з контролем  $\omega_2$  є оптимальним. ☒<sup>1</sup>

З наведеного прикладу видно, що врахування помилок різних типів дає можливість управляти надійністю та знизити ризик експлуатації кінцевого результату, що отримується на виході АП. У зв'язку з цим виникає зацікавленість в розробці теорії проектування АП з врахуванням помилок різних типів.

## 2.2. Моделі надійності операторів та логічних умов

В теорії надійності АП використовуються робочий оператор, оператор доробки, оператор оновлення та оператор тотожності [71, 79].

*Робочим оператором* називається такий оператор  $A$ , при виконанні якого помилки лише вносяться в АП, але не виявляються та не усуваються.

<sup>1</sup> Тут і надалі символом ☒ позначено кінець прикладу

При виконанні робочого оператора можливі такі події: 1 – робочий оператор виконаний безпомилково;  $0_j$  – робочий оператор виконаний з помилкою  $j$ -го типу,  $j = \overline{1, m}$ . На рис. 2.1 зображено граф переходів між станами робочого оператора. Припускається, що переходи з одних типів помилок на інші не допускаються. На рис. 1.3 через  $p_A^1$  позначено ймовірність правильного виконання робочого оператора  $A$ , а через  $p_A^{0_j}$  ймовірність внесення помилки  $j$ -го типу при виконанні оператора  $A$ , причому  $p_A^1 + \sum_{j=1}^m p_A^{0_j} = 1$ ,  $j = \overline{1, m}$ .

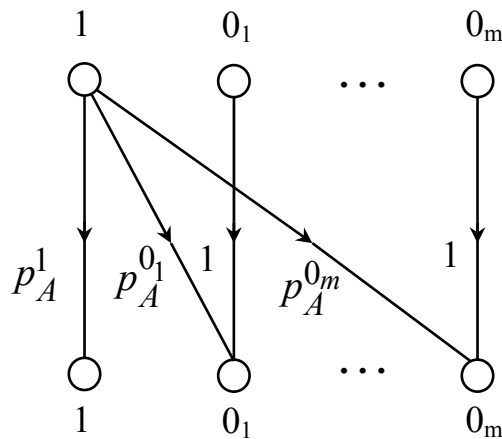


Рис. 2.1. Граф переходів при виконанні робочого оператора  $A$

Використовуючи граф переходів (рис. 2.1), модель надійності робочого оператора представимо такою матрицею:

$$\mathbf{P}_A = \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}. \quad (2.2)$$

*Оператором доробки* називається такий оператор  $U$ , при виконанні якого помилки, що виникають в АП, усуваються.

Будемо вважати, що при доробці додаткові помилки не вносяться. При виконанні оператора доробки можливі такі події: 1 – оператор доробки виконаний безпомилково;  $0_j$  – при доробці помилка  $j$ -го типу не усунена,  $j = \overline{1, m}$ .

На рис. 2.2 зображено граф переходів оператора доробки, де через  $v_U^1$  ( $v_U^{0_j}$ ) позначено ймовірність усунення (не усунення) помилки

$j$ -го типу при виконанні оператора доробки  $U$ , причому  $v_U^{1j} + v_U^{0j} = 1$ ,  $j = \overline{1, m}$ .

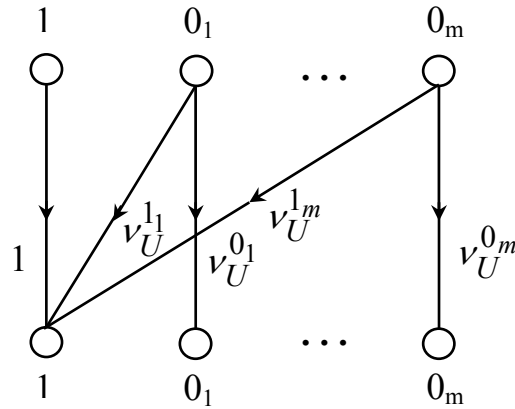


Рис. 2.2. Граф переходів при виконанні оператора доробки  $U$

За графом переходів з рис. 2.2 модель надійності оператора доробки представимо такою матрицею:

$$\mathbf{P}_U = \begin{pmatrix} 1 & 0 & \dots & 0 \\ v_U^{11} & v_U^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_U^{1m} & 0 & \dots & v_U^{0m} \end{pmatrix}. \quad (2.3)$$

На рис. 2.3. зображено граф переходів оператора доробки  $R$ , при виконанні якого вносяться додаткові помилки. На цьому рисунку через  $v_R^1$  позначено ймовірність правильного виконання оператора доробки  $R$ , а через  $v_R^{01j}$  – ймовірність внесення помилки  $j$ -го типу при доробці  $R$ , причому  $v_R^1 + \sum_{j=1}^m v_R^{01j} = 1$ ,  $j = \overline{1, m}$ .

За графом переходів з рис. 2.3 модель надійності оператора доробки  $R$  представимо такою матрицею:

$$\mathbf{P}_R = \begin{pmatrix} v_R^1 & v_R^{011} & \dots & v_R^{01m} \\ v_R^{11} & v_R^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_R^{1m} & 0 & \dots & v_R^{0m} \end{pmatrix}. \quad (2.4)$$

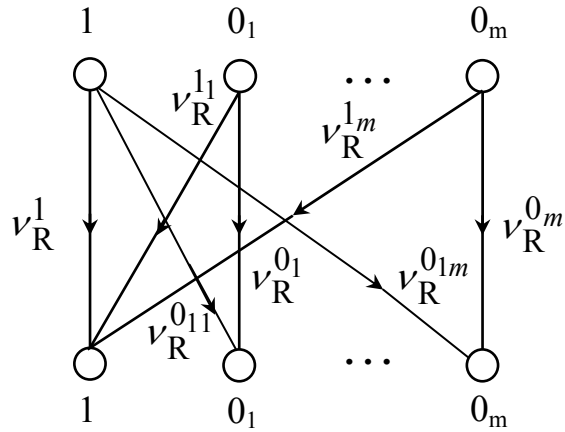


Рис. 2.3. Граф переходів при виконанні оператора доробки  $R$

*Оператором оновлення* називається такий оператор  $Z$ , який призначений для заміни робочого оператора, в якому виявлені помилки, на новий.

При виконанні оновлення  $Z$  з ймовірністю 1 відбувається перехід з будь-якого стану в стан відсутності помилок (рис. 2.4).

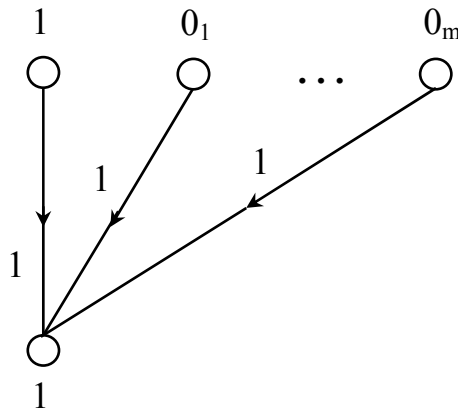


Рис. 2.4. Граф переходів при виконанні оператора оновлення  $Z$

За графом переходів з рис. 2.4 модель надійності оператора оновлення представимо такою матрицею:

$$\mathbf{P}_Z = \begin{pmatrix} 1 & 2 & \dots & m+1 \\ 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & \dots & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ \dots \\ m+1 \end{matrix} \quad (2.5)$$

Оператор оновлення  $Z$  вводиться в АП, коли при виявленні помилок відбувається повна переробка виконаного робочого оператора.

*Тотожним оператором* називається оператор  $E$ , який не змінює стан АП.

В реальних АП цей оператор відсутній, але його вводять для моделювання АП засобами мови алгоритмічних алгебр [18]. На рис. 2.5 зображено тотожний оператор  $E$  у вигляді графа.

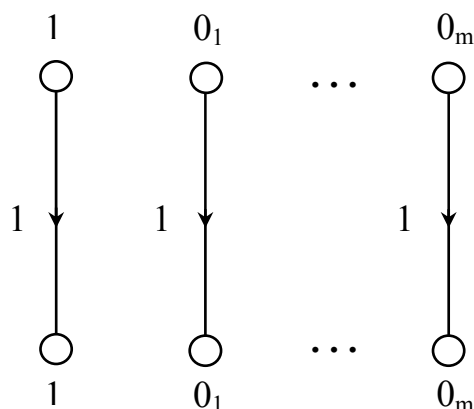


Рис. 2.5. Тотожний оператор  $E$

За графом з рис. 2.4 модель надійності оператора тотожності представимо такою матрицею:

$$\mathbf{P}_E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}. \quad (2.6)$$

Для підвищення надійності алгоритму в процес функціонування вводять контроль.

*Контролем* називається логічна умова  $\omega$ , яка вводиться в АП для виявлення допущених помилок. При виконанні контролю можливі такі події:

11 (10) – відсутність помилок ідентифіковано контролем правильно (неправильно);

00<sub>j</sub> (01<sub>j</sub>) – наявність помилки  $j$ -го типу ідентифіковано контролем правильно (неправильно).

На рис. 2.6 зображено граф переходів при контролі з врахуванням помилок першого та другого роду. На цьому рисунку через  $k_\omega^{11}$  ( $k_\omega^{10}$ ) позначено ймовірність того, що відсутність помилок ідентифіковано правильно (неправильно), а через  $k_\omega^{01j}$  ( $k_\omega^{00j}$ ) ймовірність пропуску (виявлення) помилки  $j$ -го типу, причому  $k_\omega^{01j} + k_\omega^{00j} = 1$ ,  $j = \overline{1, m}$ .

На рис. 2.6 ймовірності  $p_\omega^1$  та  $p_\omega^0$  формуються до перевірки умови  $\omega$  та необхідні для прийняття того чи іншого рішення при керу-

ванні процесом функціонування АП (рішення «1» або рішення «0»), причому:  $p_{\omega}^1$  – апріорна ймовірність відсутності помилок в АП;  $p_{\omega}^{0_j}$  – апріорна ймовірність наявності помилок  $j$ -го типу,  $j = \overline{1, m}$  ( $p_{\omega}^1 + \sum_{j=1}^m p_{\omega}^{0_j} = 1$ ).

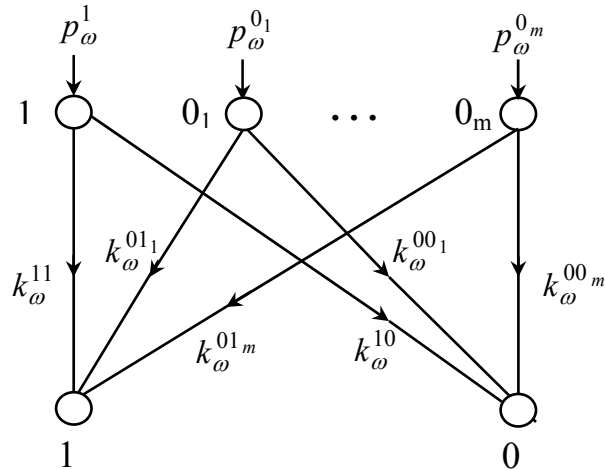


Рис. 2.6. Граф переходів при виконанні контролю  $\omega$

За графом переходу контролю  $\omega$  його модель надійності представимо такими матрицями:

$$\mathbf{K}_{\omega}^1 = \begin{pmatrix} k_{\omega}^{11} & 0 & \dots & 0 \\ 0 & k_{\omega}^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_{\omega}^{01_m} \end{pmatrix}; \quad \mathbf{K}_{\omega}^0 = \begin{pmatrix} k_{\omega}^{10} & 0 & \dots & 0 \\ 0 & k_{\omega}^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_{\omega}^{00_m} \end{pmatrix}, \quad (2.7)$$

причому  $\mathbf{K}_{\omega}^1 + \mathbf{K}_{\omega}^0 = \mathbf{I}$ , де  $\mathbf{I}$  – одинична матриця.

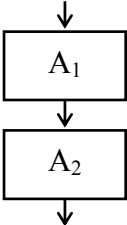
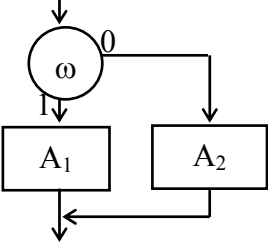
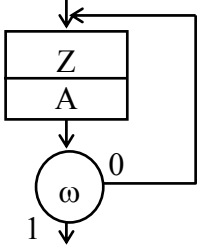
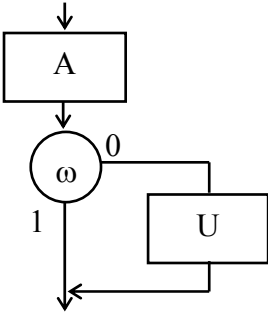
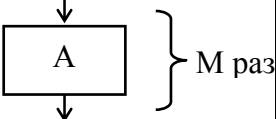
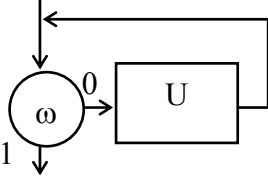
Матричне представлення операторів (2.2)–(2.6) та логічної умови (2.7) забезпечує компактність моделей прогнозування надійності багатовимірних АП.

### 2.3. Моделі надійності алгоритмічних структур

Алгоритмічна структура – це така комбінація операторів та логічних умов, для яких отриманні математичні моделі, які дозволяють замінити її єдиним оператором з еквівалентними характеристиками надійності. Згідно з теоремою про регуляризацію довільний алгоритм може бути представлений у вигляді таких алгоритмічних структур: «послідовна», « $\omega$ -диз'юнкція» та « $\omega$ -ітерація». В табл. 2.3 наведено

Таблиця 2.3

## Моделі надійності типових алгоритмічних структур

Назва	Позначення	Граф-схема	Еквівалентні характеристики
Послідовна	$B = A_1 \cdot A_2$		$\mathbf{P}_B = \mathbf{P}_{A_1} \cdot \mathbf{P}_{A_2}$ $t_B = t_{A_1} + t_{A_2}$
$\omega$ -диз'юнкція	$C = (A_1 \vee A_2)_{\omega}$		$\mathbf{P}_C = \mathbf{P}_{\omega} \cdot (\mathbf{K}_{\omega}^1 \cdot \mathbf{P}_{A_1} + \mathbf{K}_{\omega}^0 \cdot \mathbf{P}_{A_2})$ $t_C = t_{\omega} + (1-b) \cdot t_{A_1} + b \cdot t_{A_2}$ $b = p_{\omega}^1 \cdot k_{\omega}^{10} + \sum_{j=1}^m p_{\omega}^{0j} \cdot k_{\omega}^{00j}$
$\omega$ -ітерація	$D = \{A\}_{\omega}$		$\mathbf{P}_D = \mathbf{P}_Z \cdot \mathbf{P}_A \cdot (\mathbf{I} - \mathbf{K}_{\omega}^0 \cdot \mathbf{P}_Z \cdot \mathbf{P}_A)^{-1} \cdot \mathbf{K}_{\omega}^1$ $t_D = \frac{t_{\omega} + t_A}{1-b}$ $b = p_{\omega}^1 \cdot k_{\omega}^{10} + \sum_{j=1}^m p_{\omega}^{0j} \cdot k_{\omega}^{00j}$
Робота-контроль-добробка	$F = A(E \vee U)_{\omega}$		$\mathbf{P}_F = \mathbf{P}_A \cdot \mathbf{K}_{\omega}^1 + \mathbf{P}_A \cdot \mathbf{K}_{\omega}^0 \cdot \mathbf{P}_U$ $t_F = t_A + t_{\omega} + t_U \cdot b$ $b = p_{\omega}^1 \cdot k_{\omega}^{10} + \sum_{j=1}^m p_{\omega}^{0j} \cdot k_{\omega}^{00j}$
Багаторазова робота	$G = A^M$		$\mathbf{P}_G = (\mathbf{P}_A)^M$ $t_G = M \cdot t_A$
$\omega$ -добробка	$L = \{U\}_{\omega}$		$\mathbf{P}_L = \mathbf{P}_{\omega} \cdot (\mathbf{I} - \mathbf{K}_{\omega}^0 \cdot \mathbf{P}_U)^{-1} \cdot \mathbf{K}_{\omega}^1$ $t_L = t_{\omega} + b \cdot \frac{t_{\omega} + t_U}{1-b}$ $b = p_{\omega}^1 \cdot k_{\omega}^{10} + \sum_{j=1}^m p_{\omega}^{0j} \cdot k_{\omega}^{00j}$



моделі надійності цих алгоритмічних структур, а також структур «робота-контроль-доробка», «багаторазова робота» та « $\omega$ -доробка», які часто зустрічаються при проектуванні АП. В цій таблиці через  $t_A, t_U$  та  $t_\omega$  позначено час виконання робочого оператора  $A$ , оператора доробки  $U$  та контролю  $\omega$ . Використання моделей надійності (табл. 2.3) дозволяє перетворити довільний АП до єдиного робочого оператора з еквівалентними характеристиками часу виконання, ймовірності безпомилкового виконання та ймовірностей наявності помилок кожного типу.

*Приклад 2.2.* Розглянемо структуру «робота-контроль-доробка»  $F=A(E\vee U)$ . Ймовірнісно-часові характеристики робочого оператора  $A$ , оператора доробки  $U$  та контролю  $\omega$  представлені в табл. 2.4. Необхідно обчислити еквівалентні характеристики надійності цієї структури.

Таблиця 2.4

Ймовірнісно-часові характеристики елементів структури  
«робота-контроль-доробка»

Робочий оператор $A$	Оператор доробки $U$	Контроль $\omega$
$\mathbf{P}_A = \begin{pmatrix} 0,9 & 0,03 & 0,07 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\mathbf{P}_U = \begin{pmatrix} 1 & 0 & 0 \\ 0,9 & 0,1 & 0 \\ 0,8 & 0 & 0,2 \end{pmatrix}$	$\mathbf{K}_\omega^1 = \begin{pmatrix} 0,9 & 0 & 0 \\ 0 & 0,02 & 0 \\ 0 & 0 & 0,05 \end{pmatrix}$
$t_A = 4$	$t_U = 8$	$t_\omega = 2$

Еквівалентні характеристики надійності структури «робота-контроль-доробка» розрахуємо за формулами з табл. 2.3

$$\begin{aligned} \mathbf{P}_F &= \mathbf{P}_A \cdot \mathbf{K}_\omega^1 + \mathbf{P}_A \cdot \mathbf{K}_\omega^0 \cdot \mathbf{P}_U = \begin{pmatrix} 0,9 & 0,03 & 0,07 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left[ \begin{pmatrix} 0,9 & 0 & 0 \\ 0 & 0,02 & 0 \\ 0 & 0 & 0,05 \end{pmatrix} + \right. \\ &+ \left. \left( \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0,9 & 0 & 0 \\ 0 & 0,02 & 0 \\ 0 & 0 & 0,05 \end{pmatrix} \right) \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0,9 & 0,1 & 0 \\ 0,8 & 0 & 0,2 \end{pmatrix} \right] = \begin{pmatrix} 0,979 & 0,004 & 0,017 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ t_F &= t_A + t_\omega + t_U \cdot \left( p_A^1 \cdot k_\omega^{10} + \sum_{j=1}^m p_A^{0j} \cdot k_\omega^{00j} \right) = 4 + 2 + 8 \cdot (0,9 \cdot 0,1 + \\ &+ 0,03 \cdot 0,98 + 0,07 \cdot 0,95) = 7,5. \end{aligned}$$

Таким чином середній час та ймовірність безпомилкового виконання структури «робота-контроль-доробка» складає 7,5 та 0,979, а ймовірності появи помилок 1-го та 2-го типу складає 0,004 та 0,017. ☒

## 2.4. Алгоритм моделювання надійності

Розрахунок показників надійності багатовимірних АП розраховується за таким алгоритмом:

*⟨Задати структуру АП у вигляді суперпозиції  $M$  алгоритмічних структур ⟩;*

*⟨Встановити лічильник алгоритмічних структур:  $i:=1$  ⟩;*

**While** *⟨Виконується умова  $i \leq M$  ⟩ do*

**If** *⟨  $i$ -та структура «послідовна»? ⟩*

*⟨ Замінити  $i$ -ту структуру єдиним оператором за формулою структури «послідовна» з табл.2.3 ⟩;*

**End;**

**If** *⟨  $i$ -та структура « $\omega$ -диз'юнкція»? ⟩*

*⟨Замінити  $i$ -ту структуру єдиним оператором за формулою структури « $\omega$ -диз'юнкція» з табл. 2.3⟩;*

**End;**

**If** *⟨  $i$ -та структура « $\omega$ -ітерація»? ⟩*

*⟨Замінити  $i$ -ту структуру єдиним оператором за формулою структури « $\omega$ -ітерація» з табл. 2.3 ⟩;*

**End;**

**If** *⟨  $i$ -та структура «робота-контроль-доробка»? ⟩*

*⟨Замінити  $i$ -ту структуру єдиним оператором за формулою структури «робота-контроль-доробка» з табл. 2.3 ⟩;*

**End;**

**If** *⟨  $i$ -та структура «багаторазова робота»? ⟩*

*⟨Замінити  $i$ -ту структуру єдиним оператором за формулою структури «багаторазова робота» з табл. 2.3 ⟩;*

**End;**

**If** *⟨  $i$ -та структура « $\omega$ -доробка»? ⟩*

*⟨Замінити  $i$ -ту структуру єдиним оператором за формулою структури « $\omega$ -доробка» з табл. 2.3 ⟩;*

**End;**

**End.**

### 3. ФОРМАЛІЗАЦІЯ ЗАДАЧ ОПТИМІЗАЦІЇ НАДІЙНОСТІ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

Формулюються постановки задач оптимізації АП, що враховують різноманітні способи управління надійністю. Пропонується загальна постановка задачі векторної оптимізації АП та аналізуються способи її скаляризації за допомогою функції ризику, цільового програмування, умовної оптимізації та їх поєднання. Пропонується підхід до синтезу окремих задач оптимізації з загальних задач, які віддзеркалюють практичні потреби управління надійністю АП.

#### 3.1. Формалізація керованих змінних для задач оптимізації надійності

З інженерної точки зору найбільш доцільними шляхами управління надійністю АП є варіювання: а) переліком та кількістю контрольних процедур; б) варіантами реалізацій операторів та логічних умов; в) факторами, що впливають на характеристики надійності операторів та логічних умов. Шляхи (а) та (б) можна інтерпретувати як структурну оптимізацію АП, а (в) – як параметричну оптимізацію. Нижче формалізуються шляхи варіювання надійності за допомогою векторів керованих змінних.

Нехай АП складається з  $n$  потенційно-контрольованих операторів. Тоді перелік контрольних процедур опишемо таким вектором

$$\mathbf{X}_c = (x_{c1}, x_{c2}, \dots, x_{cn}),$$

в якому координата  $x_{ci}$  вказує на наявність ( $x_{ci} = 1$ ) або відсутність ( $x_{ci} = 0$ ) контролю після  $i$ -го оператора,  $i = \overline{1, n}$ .

Кратність контрольних процедур теж можна задати вектором  $\mathbf{X}_c$  за умови, що його координати прийматимуть цілочисельні значення 0, 1, 2, ....

В задачах оптимізації АП крім покращувальних підстановок з [79] можуть використовуватися і ітеративні структури «робота–контроль–оновлення» або «робота–контроль–добробка» з пам'яттю, в яких число зворотних зв'язків обмежено. В цьому випадку координати вектора  $\mathbf{X}_c$  прийматимуть цілочисельні значення 0, 1, 2, ..., які задають граничну кількість перевірок. Нульове значення керованої змінної вказує на відсутність контролю.

Нехай АП складається з  $h$  операторів та логічних умов, які можна реалізувати різними способами. Тоді варіант АП опишемо таким вектором:

$$\mathbf{X}_a = (x_{a1}, x_{a2}, \dots, x_{ah}),$$

де  $x_{ai} \in \{1, 2, \dots\}$  – вказує на номер способу альтернативної реалізації  $i$ -ї операції,  $i = \overline{1, h}$ .

Позначимо через  $x_{fi}$  –  $i$ -й фактор впливу на надійність елемента АП, значення якого можна змінювати. Тоді варіант АП опишемо таким вектором:

$$\mathbf{X}_f = (x_{f1}, x_{f2}, \dots, x_{fk}),$$

де  $k$  – кількість факторів впливу.

Формально надійність АП оптимізується зміною координат векторів  $\mathbf{X}_c$ ,  $\mathbf{X}_a$  та  $\mathbf{X}_f$ . Можливі варіанти формування керованих змінних з цих векторів наведені в табл. 3.1. Наприклад, для першого варіанта – для задачі оптимальної розстановки контрольних точок – керовані змінні задаються вектором  $\mathbf{X}_c$  з бінарними значеннями координат. Всього існує 11 варіантів формування вектора  $\mathbf{X}$  керованих змінних задачі оптимізації надійності АП з векторів  $\mathbf{X}_c$ ,  $\mathbf{X}_a$  та  $\mathbf{X}_f$ . З цих 11 задач оптимізації 5 розглядалися раніше, а 6 є новими.

Таблиця 3.1

Класифікація задач оптимізації надійності АП за переліком керованих змінних ( $B$  – бінарні значення,  $M$  – цілочисельні значення,  $N$  – не застосовується,  $Y$  – застосовується)

Новизна	№	Керовані змінні			Назва задачі
		$X_c$	$X_a$	$X_f$	
Відомі задачі	1	$B$	$N$	$N$	Розстановка контрольних точок
	2	$M$	$N$	$N$	Вибір кратності контролю
	3	$N$	$Y$	$N$	Вибір способів реалізації елементів
	4	$B$	$Y$	$N$	Загальна задача структурної оптимізації АП
	5	$M$	$Y$	$N$	
Нові задачі	6	$N$	$N$	$Y$	Параметрична оптимізація АП
	7	$N$	$Y$	$Y$	Структурно-параметрична оптимізація АП
	8	$B$	$N$	$Y$	
	9	$M$	$N$	$Y$	
	10	$B$	$Y$	$Y$	
	11	$M$	$Y$	$Y$	

### 3.2. Показники надійності, які враховуються при оптимізації

При оптимізації надійності АП пропонується враховувати такі показники:

$p^1$  – ймовірність безпомилкового виконання АП;

$p^{0j}$  – ймовірність появи помилки  $j$ -го типу на виході АП,  $j = \overline{1, m}$ ;

$I$  – ймовірність переривання АП;

$C$  – вартість виконання АП;

$T$  – тривалість виконання АП;

$S$  – загальна кількість допоміжних (контрольно–добробчих) процедур в АП.

Показники  $p^1$ ,  $C$  та  $T$  застосовували при оптимізації АП і раніше. Використання показників  $p^{0j}$ ,  $I$  та  $S$  при оптимізації надійності АП пропонується цією роботою. Показники  $p^{0j}$ ,  $j = \overline{1, m}$  застосовуються при моделюванні надійності багатовимірних АП. Показник  $I$  доцільно застосовувати при оптимізації АП, при виконанні яких можливі переривання через вичерпання ресурсів. Прикладами обмежених ресурсів є число запасних заготовок в технологічному процесі виробництва продукції, кількість спроб введення пін-коду при отриманні готівки через банкомат або ліміт часу на виконання курсової роботи студентом університету. Показник  $S$  доцільно враховувати при оптимізації АП, операції якого з деяких причин не можливо перевірити в повному обсязі. Наприклад, АП з 50 операцій виконується в приміщенні, в якому одночасно можна розташувати лише 5 робочих місць контролерів.

### 3.3. Задачі багатокритеріальної оптимізації надійності

В загальному випадку оптимізацію надійності АП за критеріями  $Criteria = \{p^1, p^{0_1}, p^{0_2}, \dots, p^{0_m}, I, C, T, S\}$  сформулюємо як задачу векторної оптимізації. Вона являє собою пошук варіанта АП у вигляді деякого вектора керованих змінних  $\mathbf{X}$ , який забезпечує

$$\left. \begin{array}{l} p^1(\mathbf{X}) \rightarrow \max, \\ p^{0j}(\mathbf{X}) \rightarrow \min, \quad j = \overline{1, m} \\ I(\mathbf{X}) \rightarrow \min, \\ C(\mathbf{X}) \rightarrow \min, \\ T(\mathbf{X}) \rightarrow \min, \\ S(\mathbf{X}) \rightarrow \min. \end{array} \right\} \quad (3.1)$$

Задачі векторної оптимізації АП (3.1), які виникають на практиці, породжують Парето-множину великої потужності. Тому доцільно замість інтерактивного перебору паретівських альтернатив, скаляризувати задачу векторної оптимізації АП та вирішувати її добре розробленими методами пошуку екстремуму.

### 3.4. Скаляризація задачі багатокритеріальної оптимізації надійності

Можливі 4 підходи скаляризації задачі багатокритеріальної оптимізації надійності АП. Перший підхід базується на мінімізації середніх сумарних витрат на основі функції ризику, другий – ідеях цільового програмування, третій – на ідеях умовної оптимізації, а четвертий поєднує другий та третій підходи.

#### 3.4.1. Застосування функції ризику

Відповідно до [79] задачу оптимізації АП можна звести до мінімізації середніх сумарних витрат

$$C(\mathbf{X}) + C_e(\mathbf{X}) \rightarrow \min, \quad (3.2)$$

де  $C_e(\mathbf{X})$  – майбутні експлуатаційні витрати, які викличе результат виконання АП  $\mathbf{X}$ .

Експлуатаційні витрати з ймовірностями помилок на виході АП пов'язані деякою спадною функцією

$$C_E(\mathbf{X}) = f(p^{0_1}(\mathbf{X}), p^{0_2}(\mathbf{X}), \dots, p^{0_m}(\mathbf{X})).$$

Для практичних задач цю залежність зазвичай апроксимують лінійно через функцію ризику. В цьому випадку (3.2) перетвориться в таку задачу:

$$C(\mathbf{X}) + \sum_{j=1}^m p^{0_j}(\mathbf{X}) \cdot R_j \rightarrow \min, \quad (3.3)$$

де  $R_j$  – майбутні витрати через наявність помилки  $j$ -го типу на виході АП,  $j = \overline{1, m}$ .

При відсутності достовірних даних про коефіцієнти функції ризику (що часто буває на етапі проектування АП) оптимізувати за постановкою (3.3) неможливо. Крім того, в постановці (3.3) не враховані показники  $I$ ,  $T$  та  $S$ . Тому практичне застосування скаляризації за цим підходом при оптимізації надійності АП є вкрай обмеженим.

### 3.4.2. Застосування цільового програмування

Визначимо мету оптимізації, тобто бажані значення показників  $p^1, p^{0j}$  ( $j = \overline{1, m}$ ),  $I, C, T$  та  $S$ . Позначимо їх  $\dot{p}^1, \dot{p}^{0j}, \dot{I}, \dot{C}, \dot{T}$  та  $\dot{S}$ . Тоді задачу векторної оптимізації (3.1) запишемо у такій скалярній формі:

$$\Delta(\mathbf{X}) = \sum_{K \in \text{Criteria}} b_K \cdot \Delta_K^- + \sum_{K \in \text{Criteria}} g_K \cdot \Delta_K^+ \rightarrow \min, \quad (3.4)$$

де  $\Delta(\mathbf{X})$  – відхилення від поставленої мети АП, який задано вектором  $\mathbf{X}$ ;

$g_K$  – штраф за перевищення на одиницю бажаного значення  $\dot{K}$  за  $K$ -им критерієм;

$b_K$  – штраф за відхилення на одиницю в меншу сторону від бажаного значення  $\dot{K}$  за  $K$ -м критерієм;

$$\Delta_K^- = \begin{cases} \dot{K} - K(\mathbf{X}), & \text{якщо } \dot{K} - K(\mathbf{X}) > 0 \\ 0, & \text{інакше} \end{cases} \quad \text{– відхилення в меншу сторону від бажаного значення за } K\text{-м критерієм};$$

рону від бажаного значення за  $K$ -м критерієм;

$$\Delta_K^+ = \begin{cases} K(\mathbf{X}) - \dot{K}, & \text{якщо } K(\mathbf{X}) - \dot{K} > 0 \\ 0, & \text{інакше} \end{cases} \quad \text{– відхилення в більшу сторону від бажаного значення за } K\text{-м критерієм.}$$

рону від бажаного значення за  $K$ -м критерієм.

Для критеріїв, які потрібно максимізувати,  $g_K > 0$  та  $b_K \leq 0$ . В цьому випадку  $abs(b_K)$  можна розглядати як премію за перевищення встановленого рівня  $\dot{K}$  за  $K$ -м критерієм. Для критеріїв, які потрібно мінімізувати,  $g_K \leq 0$  та  $b_K > 0$ . В цьому випадку  $abs(g_K)$  ототожнюється з премією за недобір за  $K$ -м критерієм.

Порахуємо кількість окремих постановок задач оптимізації АП, які можна синтезувати з загальної постановки (3.4) вилученням одного або декількох критеріїв. На загал, існує  $2^6 = 64$  комбінацій критеріїв з 6-ти елементної множини Criteria. В задачі оптимізації надійності АП має враховуватися хоча б один із ймовірнісних критеріїв ( $p^1$  або  $p^{0j}$ ) та хоча б один з ресурсних критеріїв ( $C, T$  або  $S$ ). З 64 можливих варіантів існує 42 коректні та 22 недопустимі комбінації критеріїв (табл. 3.2). З 42 коректних комбінацій критеріїв половина містить  $S$ , тобто стосується задач оптимізації контролів. В цій 21 задачі оптимізації керовані змінні обов'язково включають координати вектора  $X_c$ , тому кількість коректних комбінацій керованих змінних становитиме 8 (див. табл. 3.1). Відповідно, кількість коректних постановок з крите-

рієм  $S$  становитиме  $N_S = 21 \cdot 8 = 168$ . Для решти задач оптимізації АП існує 11 комбінацій керованих змінних (див. табл. 3.1). Відповідно, кількість коректних постановок задач, які не передбачають зміну числа контролей, становитиме  $N_{NS} = 21 \cdot 11 = 231$ . Таким чином, за формулою (3.4) можна синтезувати  $N_1 = N_S + N_{NS} = 168 + 231 = 399$  окремих задач оптимізації надійності АП. З них на оптимізацію контролів припадає  $N_{1C} = 42 \cdot 8 = 336$  задач.

Таблиця 3.2

Некоректні комбінації критеріїв при оптимізації надійності АП за (3.4)  
( $N$  – відсутній,  $Y$  – наявний,  $A$  – довільне значення)

Критерії						Кількість комбінації
$p^1$	$p^{0_j}$	$I$	$C$	$T$	$S$	
$N$	$N$	$A$	$A$	$A$	$A$	16
$Y$	$Y$	$A$	$N$	$N$	$N$	2
$Y$	$N$	$A$	$N$	$N$	$N$	2
$N$	$Y$	$A$	$N$	$N$	$N$	2
Всього некоректних комбінації						22

### 3.4.3. Застосування умовної оптимізації

Серед показників  $p^1$ ,  $p^{0_j}$  ( $j = \overline{1, m}$ ),  $I$ ,  $C$ ,  $T$  та  $S$  оберемо один найголовніший та визначимо його як критерій оптимізації. Решта показників відіграватимуть роль обмежень. Можливі постановки таких задач оптимізації АП зведені в табл. 3.3

Таблиця 3.3

Класифікація задач умовної оптимізації надійності АП  
( $G$  – критерій оптимальності,  $C$  – обмеження,  $N$  – не застосовується)

Група задач	Показники алгоритмічного процесу						Кількість задач
	$p^1$	$p^{0_j}$	$I$	$C$	$T$	$S$	
1	$G$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	28
2	$C$ або $N$	$G$ та $C$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	28
3	$C$ або $N$	$C$ або $N$	$G$	$C$ або $N$	$C$ або $N$	$C$ або $N$	21
4	$C$ або $N$	$C$ або $N$	$C$ або $N$	$G$	$C$ або $N$	$C$ або $N$	24
5	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$G$	$C$ або $N$	24
6	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$C$ або $N$	$G$	24



Розглянемо *першу групу* задач оптимізації з максимізації  $p^1$ . Якщо враховується лише обмеження за  $C$  або за  $T$ , тоді це відомі задачі оптимізації надійності в зворотній постановці:

$$\begin{cases} p^1(\mathbf{X}) \rightarrow \max \\ C(\mathbf{X}) \leq C^* \end{cases}, \quad (3.5)$$

$$\begin{cases} p^1(\mathbf{X}) \rightarrow \max \\ T(\mathbf{X}) \leq T^* \end{cases}, \quad (3.6)$$

де  $C^*$  та  $T^*$  – максимально допустимі вартість та тривалість АП.

При оптимізації багатовимірних АП в постановки (3.5)–(3.6) додамо обмеження за допустимим рівнем помилок кожного типу на виході АП. Тоді задачі оптимізації матимуть вигляд

$$\begin{cases} p^1(\mathbf{X}) \rightarrow \max \\ C(\mathbf{X}) \leq C^* \\ p^{0j}(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{cases}; \quad (3.7)$$

$$\begin{cases} p^1(\mathbf{X}) \rightarrow \max \\ T(\mathbf{X}) \leq T^* \\ p^{0j}(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{cases}, \quad (3.8)$$

де  $p^{0j}$  – максимально допустима ймовірність появи помилки  $j$ -го типу на виході АП,  $j = \overline{1, m}$ .

Аналогічно до (3.5)–(3.8) поставимо задачі оптимізації надійності з іншими комбінаціями обмежень. Порахуємо кількість можливих окремих постановок задач оптимізації. Для п'яти показників існує  $2^5 = 32$  комбінацій різних обмежень. Врахуємо, що задача оптимізації має містити хоча б одне обмеження, тому кількість допустимих комбінацій зменшиться на 1. Крім того, в задачі оптимізації має бути хоча б одне ресурсне обмеження ( $C$ ,  $T$  або  $S$ ). Відповідно, недопустимими будуть задачі оптимізації надійності з обмеженнями лише за  $p^{0j}$ , за  $I$  або за  $I$  та  $p^{0j}$  разом. Таким чином, число коректних постановок задач зменшиться ще на 3 і становить  $n_1 = 28$ . Ці постановки задач зведені в табл. 3.4. Серед них новими є 26 постановок задач. В табл. 3.4 використовуються такі позначення:  $I^*$  – максимально допустима ймовірність переривання АП;  $S^*$  – максимально допустима кількість контролів АП.

Таблиця 3.4

Постановки задач максимізації ймовірності правильного виконання  
АП ( $C$  – обмеження,  $N$  – не застосовується)

№	$p^{0j}$	$I$	$C$	$T$	$S$	Постановка задачі
1	$N$	$N$	$N$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ S \leq S^* \end{cases}$
2	$N$	$N$	$N$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ T \leq T^* \end{cases}$
3	$N$	$N$	$N$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ T \leq T^* \text{ ма } S \leq S^* \end{cases}$
4	$N$	$N$	$C$	$N$	$N$	$\begin{cases} p^1 \rightarrow \max \\ C \leq C^* \end{cases}$
5	$N$	$N$	$C$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ C \leq C^* \text{ ма } S \leq S^* \end{cases}$
6	$N$	$N$	$C$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ T \leq T^* \text{ ма } S \leq S^* \end{cases}$
7	$N$	$N$	$C$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ C \leq C^* \text{ ма } T \leq T^* \text{ ма } S \leq S^* \end{cases}$
8	$N$	$C$	$N$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, S \leq S^* \end{cases}$
9	$N$	$C$	$N$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, T \leq T^* \end{cases}$
10	$N$	$C$	$N$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, T \leq T^*, S \leq S^* \end{cases}$
11	$N$	$C$	$C$	$N$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^* \end{cases}$
12	$N$	$C$	$C$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, S \leq S^* \end{cases}$
13	$N$	$C$	$C$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, T \leq T^* \end{cases}$
14	$N$	$C$	$C$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, T \leq T^*, S \leq S^* \end{cases}$

Продовження таблиці 3.4

№	$p^{0j}$	$I$	$C$	$T$	$S$	Постановка задачі
15	$C$	$N$	$N$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ S \leq S^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
16	$C$	$N$	$N$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ T \leq T^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
17	$C$	$N$	$N$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ S \leq S^*, T \leq T^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
18	$C$	$N$	$C$	$N$	$N$	$\begin{cases} p^1 \rightarrow \max \\ C \leq C^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
19	$C$	$N$	$C$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ S \leq S^*, C \leq C^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
20	$C$	$N$	$C$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ T \leq T^*, C \leq C^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
21	$C$	$N$	$C$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ S \leq S^*, C \leq C^*, T \leq T^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
22	$C$	$C$	$N$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, S \leq S^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
23	$C$	$C$	$N$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, T \leq T^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
24	$C$	$C$	$N$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, T \leq T^*, S \leq S^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
25	$C$	$C$	$C$	$N$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
26	$C$	$C$	$C$	$N$	$C$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, S \leq S^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
27	$C$	$C$	$C$	$C$	$N$	$\begin{cases} p^1 \rightarrow \max \\ I \leq I^*, C \leq C^*, T \leq T^*, p^{0j} \leq q_j, j = \overline{1, m} \end{cases}$
28	$C$	$C$	$C$	$C$	$C$	$\begin{cases} p^1 \rightarrow \max \\ p^{0j} \leq q_j, j = \overline{1, m}, \\ I \leq I^*, C \leq C^*, T \leq T^*, S \leq S^* \end{cases}$

Постановки задач з *другої групи* стосуються оптимізації багатовимірних процесів, в яких доцільно мінімізувати ймовірність появи найбільш небезпечної помилки на виході АП. Щодо помилок інших типів, то ймовірності їх появи обмежимо зверху. Кількість різних коректних постановок задач цієї групи становить  $n_2 = 28$ . Їх синтезуємо так само, як і для першої групи, тобто за табл. 3.4. Усі постановки задач цієї групи є новими.

В задачах *третьої групи* мінімізуємо ймовірність переривання АП. Ці задачі мають включати хоча б одне ресурсне обмеження ( $C$ ,  $T$  або  $S$ ) та хоча б одне обмеження з рівня безпомилковості ( $p^1$  або  $p^{0j}$ ). Таким чином, кількість різних коректних постановок задач цієї групи становить  $n_3 = (2^3 - 1) \cdot (2^2 - 1) = 21$ . Усі постановки задач цієї групи є новими.

В задачах *четвертої та п'ятої груп* мінімізуємо вартість виконання АП та його тривалість відповідно. Якщо враховується лише обмеження за  $p^1$ , тоді це відомі задачі оптимізації надійності в прямій постановці

$$\begin{cases} C(\mathbf{X}) \rightarrow \min \\ p^1(\mathbf{X}) \geq P^* \end{cases} \quad \text{або} \quad \begin{cases} T(\mathbf{X}) \rightarrow \min \\ p^1(\mathbf{X}) \geq P^* \end{cases}, \quad (3.9)$$

де  $P^*$  – мінімально допустима ймовірність виконання АП без помилок на виході.

При оптимізації багатовимірних АП в постановки (3.9) додамо обмеження за допустимим рівнем помилок кожного типу на виході АП. Тоді задачі оптимізації (3.9) матимуть вигляд

$$\begin{cases} C(\mathbf{X}) \rightarrow \min \\ p^1(\mathbf{X}) \geq P^* \\ p^{0j}(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{cases} \quad \text{або} \quad \begin{cases} T(\mathbf{X}) \rightarrow \min \\ p^1(\mathbf{X}) \geq P^* \\ p^{0j}(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{cases}. \quad (3.10)$$

Постановки задач четвертої та п'ятої груп мають містити хоча б одне обмеження з рівня безпомилковості ( $p^1$  або  $p^{0j}$ ). Тому загальна кількість коректних постановок в кожній групі становить  $n_4 = n_5 = 2^3 \cdot (2^2 - 1) = 24$ . Серед них 22 постановки є новими.

В задачах *шостої групи* мінімізуємо кількість контрольних процедур. Коректні постановки цих задач мають містити хоча б одне обмеження з рівня безпомилковості ( $p^1$  або  $p^{0j}$ ). Таким чином, кількість різних коректних постановок задач цієї групи становить

$n_6 = 2^3 \cdot (2^2 - 1) = 24$ . Оскільки критерій оптимальності набуває дискретних значень, то можливі декілька оптимумів. Для формалізації прийняття рішення з декількох оптимумів в постановку задачі слід вести правило вибору кращого варіанта. Найпростіше це зробити шляхом визначення пріоритетного показника з множини  $\{p^1, p^{0j}, I, C, T\}$ . Наприклад, якщо за пріоритетний обрано показник  $C$ , тоді серед варіантів АП, які задовольняють усі обмеження та містять мінімальне число контролів, як рішення обираємо альтернативу мінімальної вартості. Усі постановки задач цієї групи є новими. Таким чином, комбінуючи різні показники АП в критерії та обмеження, можна синтезувати  $n_1 + n_2 + n_3 + n_4 + n_5 + n_6 = 28 + 28 + 21 + 24 + 24 + 24 = 149$  коректних варіантів. Враховуючи, що керовані змінні в кожній задачі оптимізації можуть бути скомбіновані з координат векторів  $X_c$ ,  $X_a$  та  $X_f$  (див. табл. 3.1), отримуємо, що загальна кількість коректних задач оптимізації надійності становить  $N_2 = 1567$ . Процес розрахунку цієї кількості задач зведено в табл. 3.5. В ній враховано, що при мінімізації числа контролів керовані змінні обов'язково включають координати вектора  $X_c$ . Тому в передостанньому рядку табл. 3.5 кількість комбінацій керованих змінних дорівнює 8.

Таблиця 3.5

Розрахунок кількості коректних постановок задач умовної оптимізації надійності АП

Цільова функція	Кількість комбінацій обмежень з табл. 3.2	Кількість комбінацій керованих змінних з табл. 3.1	Кількість постановок задач
$p^1 \rightarrow \max$	28	11	$28 \times 11 = 308$
$p^{0j} \rightarrow \min$	28	11	$28 \times 11 = 308$
$I \rightarrow \min$	21	11	$21 \times 11 = 231$
$C \rightarrow \min$	24	11	$24 \times 11 = 264$
$T \rightarrow \min$	24	11	$24 \times 11 = 264$
$S \rightarrow \min$	24	8	$28 \times 8 = 192$
Всього коректних постановок задач умовної оптимізації			$N_2 = 1567$

### 3.4.4. Спільне застосування умовної оптимізації та цільового програмування

В задачу оптимізації (3.4) додамо обмеження з порогових значень критеріїв. В загальному випадку задача (3.4) перетвориться на таку:

$$\left. \begin{aligned} \Delta(\mathbf{X}) &= \sum_{K \in \text{Criteria}} b_K \cdot \Delta_K^- + \sum_{K \in \text{Criteria}} g_K \cdot \Delta_K^+ \rightarrow \min \\ \text{за умов } p^1(\mathbf{X}) &\geq P^* \\ p^{0j}(\mathbf{X}) &\leq q_j, \quad j = \overline{1, m} \\ I(\mathbf{X}) &\leq I^* \\ C(\mathbf{X}) &\leq C^* \\ T(\mathbf{X}) &\leq T^* \\ S(\mathbf{X}) &\leq S^* \end{aligned} \right\} \quad (3.11)$$

Окремі задачі оптимізації, що ґрунтуються на (3.11), матимуть від 1 до 6 обмежень. Для задач оптимізації контролів кількість можливих комбінацій обмежень становить  $O_C = 2^6 - 1 = 63$ . В задачах оптимізації, в яких не змінюється число контролів, відсутня умова  $S(\mathbf{X}) \leq S^*$ , тому кількість можливих комбінацій обмежень становить  $O_{NC} = 2^5 - 1 = 31$ . Відповідно, кількість постановок окремих задач оптимізації дорівнює

$$N_3 = N_{1C} \cdot O_C + (N_1 - N_{1C}) \cdot O_{NC} = 336 \cdot 63 + (399 - 336) \cdot 31 = 23121.$$

Таким чином, практичні задачі оптимізації надійності АП можна поставити

$$N = N_1 + N_2 + N_3 = 399 + 1567 + 23121 = 25087$$

різними способами. В розділі 4 наводяться градієнтні алгоритми оптимізації контролів в багатовимірних АП за постановками (3.7) та (3.10) та аналізуються їх переваги та недоліки. Усі поставлені задачі оптимізації можна вирішити ГА. В розділі 5 пропонуються реалізації елементів ГА, які необхідні для розв'язання поставлених задач оптимізації та порівнюються результати оптимізації з градієнтним пошуком.

## 4. ГРАДІЄНТНА ОПТИМІЗАЦІЯ КОНТРОЛІВ В БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСАХ

Пропонуються градієнтні алгоритми оптимізації найбільш важливих в практичному плані задач оптимізації надійності АП – розподіл контрольних точок та вибір кратностей контролів. Вони узагальнюють градієнтні методи оптимізації з [79] на багатовимірні АП. Матеріали розділу базуються на публікаціях авторів [79, 86, 88, 90].

### 4.1. Задачі розподілу контрольних точок

Розглядається АП, що складається з  $n$  потенційно-контрольованих робочих операторів  $A_1(x_1), A_2(x_2), \dots, A_n(x_n)$ . Потенційно-контрольований оператор записується так:

$$A_i(x_i) = \begin{cases} A_i (E \vee U_i), & \text{якщо } x_i = 1 \\ \omega_i & , i = \overline{1, n}, \\ A_i, & \text{якщо } x_i = 0 \end{cases} \quad (4.1)$$

де  $A_i$  – робочий оператор, під час виконання якого в АП можуть бути внесені помилки різних типів,  $i = \overline{1, n}$ ;

$\omega_i$  – контроль правильності виконання робочого оператора  $A_i$ , причому  $\omega_i = 1(0)$ , якщо  $A_i$  виконана без помилок (з помилками),  $i = \overline{1, n}$ ;

$U_i$  – оператор доробки, який усуває помилки, які виявлені контролем  $\omega_i$ ,  $i = \overline{1, n}$ ;

$E$  – тотожній оператор, який використовується для фіксації результату виконання робочого оператора  $A_i$ ;

$x_i$  – умова переходу на контроль після виконання робочого оператора  $A_i$ :  $x_i = 1(0)$ , якщо  $A_i$  виконується з контролем (без контролю),  $i = \overline{1, n}$ .

Оператор (4.1) може бути виконаний з контролем  $\omega_i$  та доробкою  $U_i$  або без них,  $i = \overline{1, n}$ .

Показники надійності потенційно-контрольованого робочого оператора розраховуються так:

$$\mathbf{P}_{A_i}(x_i) = (1 - x_i) \cdot \mathbf{P}_{A_i} + x_i \cdot \mathbf{P}_{A_i} \cdot (\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i}); \quad (4.2)$$

$$c_{A_i}(x_i) = (1-x_i) \cdot c_{A_i} + x_i \cdot \left( c_{A_i} + c_{\omega_i} + c_{U_i} \cdot \left( p_{A_i}^1 \cdot k_{\omega_i}^{10} + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot k_{\omega_i}^{00j} \right) \right), \quad (4.3)$$

де  $c_{A_i}$ ,  $c_{\omega_i}$  та  $c_{U_i}$  – вартість виконання робочого оператора  $A_i$ , контролю  $\omega_i$  і доробки  $U_i$ , відповідно,  $i = \overline{1, n}$ .

Запишемо в матричній формі характеристики надійності та середню вартість багатовимірного АП так:

$$\mathbf{P}(\mathbf{X}) = \prod_{i=1, \overline{n}} \mathbf{P}_{A_i}(x_i); \quad (4.4)$$

$$C(\mathbf{X}) = \sum_{i=1, \overline{n}} c_{A_i}(x_i), \quad (4.5)$$

де  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  – вектор керованих змінних, в якому координата  $x_i$  вказує на наявність ( $x_i = 1$ ) або відсутність ( $x_i = 0$ ) контролю з доробкою після  $i$ -го робочого оператора  $A_i$ ,  $i = \overline{1, n}$ ;

$$\mathbf{P}(\mathbf{X}) = \begin{pmatrix} p^1(\mathbf{X}) & p^{01}(\mathbf{X}) & \dots & p^{0m}(\mathbf{X}) \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} - \text{характеристики надійності багатовимірного АП: } p^1(\mathbf{X}) - \text{ймовірність безпомилкового виконання; } p^{0j}(\mathbf{X}) - \text{ймовірність наявності помилки } j\text{-го типу, } j = \overline{1, m}.$$

$C(\mathbf{X})$  – середня вартість виконання АП.

Використовуючи співвідношення (4.2) – (4.5) задачі оптимізації (3.7) та (3.10), перепишемо таким чином:

*Пряма постановка:* знайти  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , щоб

$$\left. \begin{aligned} & \sum_{i=1, \overline{n}} \left[ (1-x_i) \cdot c_{A_i} + x_i \cdot \left( c_{A_i} + c_{\omega_i} + c_{U_i} \cdot \left( p_{A_i}^1 \cdot k_{\omega_i}^{10} + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot k_{\omega_i}^{00j} \right) \right) \right] \rightarrow \min \\ & \prod_{i=1, \overline{n}} \left[ (1-x_i) \cdot p_{A_i}^1 + x_i \cdot \left( p_{A_i}^1 + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right) \right] \geq P^* \\ & 1 - \prod_{i=1, \overline{n}} \left[ \left( 1 - p_{A_i}^{0j} \right) \cdot (1-x_i) + x_i \cdot \left( 1 - p_{A_i}^{0j} \cdot \left( 1 - k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right) \right) \right] \leq q_j, j = \overline{1, m} \end{aligned} \right\} \quad (4.6)$$



Зворотна постановка: знайти  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , щоб

$$\left. \begin{aligned} & \prod_{i=1, \overline{n}} \left[ (1-x_i) \cdot p_{A_i}^1 + x_i \cdot \left( p_{A_i}^1 + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right) \right] \rightarrow \max \\ & \sum_{i=1, \overline{n}} \left[ (1-x_i) \cdot c_{A_i} + x_i \cdot \left( c_{A_i} + c_{\omega_i} + c_{U_i} \cdot \left( p_{A_i}^1 \cdot k_{\omega_i}^{10} + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot k_{\omega_i}^{00j} \right) \right) \right] < C^*, \\ & 1 - \prod_{i=1, \overline{n}} \left[ \left( (1-p_{A_i}^{0j}) \cdot (1-x_i) + x_i \cdot \left( 1-p_{A_i}^{0j} \cdot \left( 1-k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right) \right) \right) \right] \leq q_j, j = \overline{1, m} \end{aligned} \right\} \quad (4.7)$$

Згідно з табл. 3.2 задача (4.6) відноситься до четвертої групи задач оптимізації надійності, а задача (4.7) – до першої групи.

#### 4.2. Задачі вибору кратностей контролів

Розглядається АП  $A_1(x_1), A_2(x_2), \dots, A_n(x_n)$ , що складається з  $n$  робочих операторів, які виконуються з багаторазовим контролем та доробкою. Такі робочі оператори записуються так:

$$A_i(x_i) = A_i ( E \vee U_i )_{\omega_i}^{x_i}, \quad i = \overline{1, n}, \quad (4.8)$$

де  $x_i$  – кратність виконання контролю  $\omega_i$  з доробкою  $U_i$  після робочого оператора  $A_i$ ,  $i = \overline{1, n}$ .

Оператор (4.8) є робочим оператором  $A_i$ , після якого  $x_i$  разів виконується контроль  $\omega_i$  та доробка  $U_i$  (якщо виявлені помилки),  $i = \overline{1, n}$ .

Припустимо, що для оператора  $A_i$  характеристики всіх контролів та операторів доробки є однаковими. Тоді показники надійності оператора  $A_i(x_i)$  розраховуються за такою ітеративною схемою:

$$\left. \begin{aligned} \mathbf{P}_{A_i}^{<x_i>} &= \mathbf{P}_{A_i}^{<x_i-1>} \cdot \left( \mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i} \right) \\ \mathbf{P}_{A_i}^{<0>} &= \mathbf{P}_{A_i} \end{aligned} \right\}, \quad (4.9)$$

$$\left. \begin{aligned} c_{A_i}^{<x_i>} &= c_{A_i}^{<x_i-1>} + c_{\omega_i} + \sigma^{<x_i>} \cdot c_{U_i} \\ c_{A_i}^{<0>} &= c_{A_i} \end{aligned} \right\}, \quad (4.10)$$

де  $\sigma^{<x>} = k_{\omega}^{10} \cdot p_A^{1<x-1>} + \sum_{j=1,m} k_{\omega}^{00j} \cdot p_A^{0j<x-1>}$  – ймовірність переходу на доробку  $U_i$  після  $x_i$ -кратного контролю.

Якщо знехтувати помилками першого роду при контролі ( $k_{\omega_i}^{10} = 0$ ), то формули (4.9) та (4.10) перетворяться в такі співвідношення:

$$\mathbf{P}_{A_i}(x_i) = \mathbf{P}_{A_i} \cdot \left( \mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i} \right)^{x_i}; \quad (4.11)$$

$$c_{A_i}(x_i) = c_{A_i} + x_i \cdot c_{\omega_i} + c_{U_i} \cdot \sum_{j=1,m} p_{A_i}^{0j} \cdot \left( 1 - \left( k_{\omega_i}^{01j} \right)^{x_i} \right). \quad (4.12)$$

Зауважимо, що при  $x_i = 1$  та  $k_{\omega_i}^{10} = 0$  співвідношення (4.11), (4.12) перетворюються в співвідношення (4.2), (4.3).

В матричній формі показники надійності багатовимірнього АП запишемо таким чином:

$$\mathbf{P}(\mathbf{X}) = \prod_{i=1,n} \mathbf{P}_{A_i} \cdot \left( \mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i} \right)^{x_i}; \quad (4.13)$$

$$C(\mathbf{X}) = \sum_{i=1,n} \left[ c_{A_i} + x_i \cdot c_{\omega_i} + c_{U_i} \cdot \sum_{j=1,m} p_{A_i}^{0j} \cdot \left( 1 - \left( k_{\omega_i}^{01j} \right)^{x_i} \right) \right], \quad (4.14)$$

де  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  – вектор керованих змінних, в якому координата  $x_i \in \{0, 1, 2, \dots\}$  позначає кратність виконання процедур контролю та доробки після робочого оператора  $A_i$ .

Використовуючи співвідношення (4.13), (4.14) задачі оптимізації (3.7) та (3.10), запишемо у вигляді таких співвідношень:

*Пряма постановка:* знайти  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , щоб

$$\left. \begin{aligned} & \sum_{i=1,n} \left[ c_{A_i} + x_i \cdot c_{\omega_i} + c_{U_i} \cdot \sum_{j=1,m} p_{A_i}^{0j} \cdot \left( 1 - \left( k_{\omega_i}^{01j} \right)^{x_i} \right) \right] \rightarrow \min, \\ & \prod_{i=1,n} \left[ p_{A_i}^1 + \sum_{j=1,m} p_{A_i}^{0j} \cdot \left( 1 - \left( 1 - k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right)^{x_i} \right) \right] \geq P^*, \\ & 1 - \prod_{i=1,n} \left[ 1 - p_{A_i}^{0j} \cdot \left( 1 - k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right)^{x_i} \right] \leq q_j, j = \overline{1, m}, \end{aligned} \right\} \quad (4.15)$$

Зворотна постановка: знайти  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , щоб

$$\left. \begin{aligned} & \prod_{i=1, \overline{n}} \left[ p_{A_i}^1 + \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot \left( 1 - \left( 1 - k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right)^{x_i} \right) \right] \rightarrow \max, \\ & \sum_{i=1, \overline{n}} \left[ c_{A_i} + x_i \cdot c_{\omega_i} + c_{U_i} \cdot \sum_{j=1, \overline{m}} p_{A_i}^{0j} \cdot \left( 1 - \left( k_{\omega_i}^{00j} \right)^{x_i} \right) \right] < C^*, \\ & 1 - \prod_{i=1, \overline{n}} \left[ 1 - p_{A_i}^{0j} \cdot \left( 1 - k_{\omega_i}^{00j} \cdot v_{U_i}^{1j} \right)^{x_i} \right] \leq q_j, \quad j = \overline{1, m}. \end{aligned} \right\} \quad (4.16)$$

Згідно з таблицею 3.2 задача (4.15) відноситься до четвертої групи задач оптимізації надійності, а задача (4.16) – до першої групи.

Поставлені задачі розподілу контрольних точок та вибору кратностей контролів є нелінійними аналогами задачі про рюкзак, тому для них не існує поліноміального алгоритму розв'язання. Це обумовлює пошук евристик та метаевристик, використання яких забезпечувало б швидке знаходження якісних, тобто близьких до оптимальних розв'язків.

Нижче пропонуються швидкі моделі оптимізації для розв'язання задач (4.6), (4.7), (4.15) та (4.16), які базуються на градієнтних та генетичних принципах. Для тестування цих моделей розроблено бібліотеку задач.

### 4.3. Бібліотека тестових задач

Бібліотека тестових задач розміщена за адресою [5]. Вона складається з чотирьох множин тестових задач: А\_4\_1, А\_4\_2, В\_4\_1 та В\_4\_2. Множини задач А\_4\_1 та А\_4\_2 використовуються для тестування моделей оптимізації задач (4.6) та (4.15), а В\_4\_1 та В\_4\_2 – для задач (4.7) та (4.16). В кожній множині задач розглядаються АП з чотирма різними типами помилок ( $m=4$ ) з кількістю робочих операторів  $n=20, 40, 60, 80, 100, 120$ . Для задач з множини А\_4\_1 та В\_4\_1 існує один важливий тип помилки. Ймовірність появи цього типу помилки на виході АП необхідно забезпечити на порядок нижче за інші. Для задач з множини А\_4\_2 та В\_4\_2 всі типи помилок мають приблизно однакову важливість. Кожна тестова задача представляється текстовим файлом, в якому записані ймовірно-вартісні характеристики операторів та логічних умов АП. Фрагмент такого текстового файла зображено на рис. 4.1. В кожному рядку цього файла записані надійнісні характеристики потенційно-контрольованого робочого оператора (для задачі розподілу контрольних точок) або робочого оператора з х-

кратним контролем та доробкою (для задачі вибору кратностей контролів). Кількість рядків у файлі відповідає кількості робочих операторів в АП.

Значення ймовірнісних характеристик операторів та логічних умов генерувалися випадково з таких діапазонів [79]:

для робочих операторів:

$$c_{A_i} \in [3, 10]; p_{A_i}^1 \in [0.95, 1]; p_{A_i}^{0j} \in [0, 0.01], j = \overline{1, m}, i = \overline{1, n};$$

для операторів доробки:

$$c_{U_i} \in [5, 25]; \nu_{U_i}^1 \in [0.97, 1], j = \overline{1, m}, i = \overline{1, n};$$

для логічних умов:

$$c_{\omega_i} \in [1, 15]; k_{\omega_i}^{11} \in [0.95, 1]; k_{\omega_i}^{00j} \in [0.3, 0.8], j = \overline{1, m}, i = \overline{1, n}.$$

Вказані діапазони відповідають статистичним даним характеристик надійності технологічних операцій виробництва радіоелектронних приладів [79].

р01	р02	р03	р04	к00_1	к00_2
0.0011417063	0.0016007447	0.0016585563	0.0024586120	0.95756274	0.928533
0.0016481515	0.0017661726	0.0020122423	0.0036736446	0.97876050	0.917440
0.0026994759	0.0003407359	0.0003645178	0.0003913842	0.94187178	0.983631
0.0003892169	0.0004782012	0.0005345238	0.0005961678	0.98141854	0.930496
0.0015054629	0.0017746358	0.0022976586	0.0021373427	0.93102822	0.913879
0.0006561727	0.0009404046	0.0008405810	0.0012437161	0.91146953	0.956642
0.0017256022	0.0017980917	0.0023677130	0.0037212471	0.95715480	0.936005
0.0014943320	0.0021357343	0.0025207144	0.0022869031	0.91409000	0.980563
0.0007023570	0.0008556469	0.0009812331	0.0010267026	0.97258000	0.902256
0.0016672837	0.0018868763	0.0026801215	0.0028745715	0.92099412	0.906034
0.0001867555	0.0002455081	0.0002567384	0.0003211804	0.98388758	0.980710
0.0005883157	0.0007579651	0.0007946398	0.0010018995	0.99507863	0.905755
0.0014121957	0.0019775139	0.0018832964	0.0023779714	0.97217290	0.924461
0.0007439558	0.0010541387	0.0010239538	0.0017987496	0.96224956	0.920754
0.0009492273	0.0010918253	0.0012900934	0.0025081284	0.95597705	0.907501
0.0005793555	0.0007616198	0.0007302167	0.0011588712	0.96610142	0.987988
0.0012868647	0.0012249499	0.0016722587	0.0024592048	0.90166149	0.937979
0.0015741706	0.0020602847	0.0026638896	0.0030140238	0.99111051	0.910680
0.0013118511	0.0016519207	0.0020159246	0.0019001430	0.93014108	0.938640
0.0017286072	0.0018201982	0.0024659712	0.0030298430	0.91010605	0.979332

Рис. 4.1. Фрагмент початкових даних тестової задачі А\_4\_2 при n=20

Постановки тестових задач A\_4\_1, A\_4\_2 та B\_4\_1, B\_4\_2 наведені в табл. 4.1 та табл. 4.2.

Таблиця 4.1

Умови тестових задач A\_4\_1 та A\_4\_2

$n$	задача	$P^*$	$q_1$	$q_2$	$q_3$	$q_4$	$C$
20	A_4_1	$\geq 0,9631$	$< 0,0066$	$< 0,0240$	$< 0,0258$	$< 0,0285$	$\rightarrow \min$
	A_4_2	$\geq 0,9561$	$< 0,0081$	$< 0,0098$	$< 0,0109$	$< 0,0159$	$\rightarrow \min$
40	A_4_1	$\geq 0,9448$	$< 0,0096$	$< 0,0404$	$< 0,0410$	$< 0,0405$	$\rightarrow \min$
	A_4_2	$\geq 0,9269$	$< 0,0158$	$< 0,0198$	$< 0,0223$	$< 0,0269$	$\rightarrow \min$
60	A_4_1	$\geq 0,9305$	$< 0,0144$	$< 0,0538$	$< 0,0575$	$< 0,0590$	$\rightarrow \min$
	A_4_2	$\geq 0,9010$	$< 0,0186$	$< 0,0222$	$< 0,0269$	$< 0,0357$	$\rightarrow \min$
80	A_4_1	$\geq 0,8670$	$< 0,0237$	$< 0,0758$	$< 0,0866$	$< 0,0964$	$\rightarrow \min$
	A_4_2	$\geq 0,8725$	$< 0,0255$	$< 0,0315$	$< 0,0355$	$< 0,0390$	$\rightarrow \min$
100	A_4_1	$\geq 0,8529$	$< 0,0261$	$< 0,0632$	$< 0,0786$	$< 0,0898$	$\rightarrow \min$
	A_4_2	$\geq 0,8450$	$< 0,0299$	$< 0,0367$	$< 0,0417$	$< 0,0478$	$\rightarrow \min$
120	A_4_1	$\geq 0,8357$	$< 0,0290$	$< 0,1005$	$< 0,1112$	$< 0,1119$	$\rightarrow \min$
	A_4_2	$\geq 0,8105$	$< 0,0369$	$< 0,0460$	$< 0,0520$	$< 0,0611$	$\rightarrow \min$

Таблиця 4.2

Умови тестових задач B\_4\_1 та B\_4\_2

$n$	задача	$C^*$	$q_1$	$q_2$	$q_3$	$q_4$	$p^1$
20	B_4_1	$\leq 226,01$	$< 0,0056$	$< 0,0240$	$< 0,0258$	$< 0,0285$	$\rightarrow \max$
	B_4_2	$\leq 179,8$	$< 0,0095$	$< 0,0124$	$< 0,0138$	$< 0,0152$	$\rightarrow \max$
40	B_4_1	$\leq 441,6$	$< 0,0106$	$< 0,0404$	$< 0,0410$	$< 0,0405$	$\rightarrow \max$
	B_4_2	$\leq 317,96$	$< 0,0150$	$< 0,0189$	$< 0,0211$	$< 0,0240$	$\rightarrow \max$
60	B_4_1	$\leq 541,01$	$< 0,0139$	$< 0,0538$	$< 0,0575$	$< 0,0590$	$\rightarrow \max$
	B_4_2	$\leq 525,8$	$< 0,0187$	$< 0,0237$	$< 0,0265$	$< 0,0300$	$\rightarrow \max$
80	B_4_1	$\leq 714,96$	$< 0,0242$	$< 0,0758$	$< 0,0866$	$< 0,0964$	$\rightarrow \max$
	B_4_2	$\leq 670,11$	$< 0,0264$	$< 0,0329$	$< 0,0372$	$< 0,0413$	$\rightarrow \max$
100	B_4_1	$\leq 979,70$	$< 0,0262$	$< 0,0897$	$< 0,1061$	$< 0,1400$	$\rightarrow \max$
	B_4_2	$\leq 829,14$	$< 0,0319$	$< 0,0398$	$< 0,0445$	$< 0,0499$	$\rightarrow \max$
120	B_4_1	$\leq 1140,1$	$< 0,0278$	$< 0,1005$	$< 0,1112$	$< 0,1119$	$\rightarrow \max$
	B_4_2	$\leq 1021,2$	$< 0,0372$	$< 0,0465$	$< 0,0521$	$< 0,0584$	$\rightarrow \max$

## 4.4. Градієнтна оптимізація надійності багатовимірних алгоритмічних процесів

### 4.4.1. Ідея градієнтного пошуку

Особливістю градієнтного пошуку є лінійна апроксимація цільової функції з метою пошуку максимуму в напрямку найбільшого зростання, тобто за градієнтом [63].

Градієнтну оптимізацію розподілу контрольних точок та вибору кратностей контролів при бінарній концепції врахування помилок запропоновано в [79]. Аналогами цих моделей є градієнтний метод оптимального резервування з теорії надійності [44, 100, 114] та пожедликий алгоритм розв'язання задачі про рюкзак в дискретній оптимізації [36, 61].

В основу градієнтної оптимізації розподілу контрольних точок та вибору кратностей контролів в [79] покладено ітераційну процедуру зміни значення однієї керованої зміни, що приводить до локального покращення градієнта контролю. Градієнтом контролю називається відношення [79]

$$\gamma_i(x_i) = \frac{\Delta p^1(\mathbf{X}, x_i)}{\Delta C(\mathbf{X}, x_i)} = \frac{p^1(\mathbf{X}, x_i = 1) - p^1(\mathbf{X}, x_i = 0)}{C(\mathbf{X}, x_i = 1) - C(\mathbf{X}, x_i = 0)} \quad (4.17)$$

в якому чисельник та знаменник відповідають приростам безпомилковості та вартості виконання АП при введенні додатково контролю з доробкою після робочого оператора  $A_i$ .

Градієнт (4.17) є аналогом критерію ефективності складних систем. Градієнт (4.17) показує відносну ефективність щодо встановлення додаткового контролю з доробкою після робочого оператора  $A_i$ . Градієнт (4.17) не враховує здатність контролю та доробки, виявляти та усувати помилки різних типів. Нижче узагальнюється співвідношення (4.17) на випадок помилок різних типів.

### 4.4.2. Градієнт контролю при багатовимірному алгоритмічному процесі

Для багатовимірних АП пропонується градієнт  $i$ -го контролю розраховувати таким чином

$$\gamma_i(x_i) = \frac{P(\mathbf{X}, x_i = x_i + 1) - P(\mathbf{X}, x_i)}{C(\mathbf{X}, x_i = x_i + 1) - C(\mathbf{X}, x_i)}, \quad (4.18)$$

де чисельник відповідає приросту характеристик надійності при введенні контролю та доробки після  $i$ -го робочого оператора  $A_i$ , а знаменник приросту вартості на виконання АП.

Приріст характеристик надійності в (4.18) запишемо так:

$$\begin{aligned}
\mathbf{P}(\mathbf{X}, x_i + 1) - \mathbf{P}(\mathbf{X}, x_i) &= \mathbf{P}_{A_i}(x_i + 1) \cdot \prod_{r=1, n, r \neq i} \mathbf{P}_{A_r}(x_r) - \mathbf{P}_{A_i}(x_i) \cdot \prod_{r=1, n, r \neq i} \mathbf{P}_{A_r}(x_r) = \\
&= \left( \mathbf{P}_{A_i} \cdot (\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i})^{x_i + 1} - \mathbf{P}_{A_i} \cdot (\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i})^{x_i} \right) \cdot \prod_{r=1, n, r \neq i} \mathbf{P}_{A_r}(x_r) = \\
&= \mathbf{P}_{A_i} \cdot (\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i})^{x_i} \cdot (\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i} - \mathbf{I}) \cdot \mathbf{P}_{A_i}^{-1}(x_i) \cdot \mathbf{P}(\mathbf{X}) = \\
&= \mathbf{P}_{A_i} \cdot \mathbf{B}_i^{x_i} \cdot (\mathbf{B}_i - \mathbf{I}) \cdot \mathbf{P}_{A_i}^{-1}(x_i) \cdot \mathbf{P}(\mathbf{X}).
\end{aligned}$$

Приріст ресурсів на виконання АП в (4.18) запишемо так:

$$\begin{aligned}
C(\mathbf{X}, x_i + 1) - C(\mathbf{X}, x_i) &= c_{A_i}(x_i + 1) + \sum_{r=1, n, r \neq i} c_{A_r}(x_r) - c_{A_i}(x_i) - \sum_{r=1, n, r \neq i} c_{A_r}(x_r) = \\
&= \left( c_{A_i} + (x_i + 1) \cdot c_{\omega_i} + c_{U_i} \cdot \sum_{j=1}^m \left( p_{A_i}^{0j} \cdot (1 - (k_{\omega_i}^{01j})^{x_i + 1}) \right) \right) - c_{A_i} + x_i \cdot c_{\omega_i} + \\
&+ c_{U_i} \sum_{j=1}^m \left( p_{A_i}^{0j} \cdot (1 - (k_{\omega_i}^{01j})^{x_i}) \right) = c_{\omega_i} + c_{U_i} \sum_{j=1, m} p_{A_i}^{0j} \cdot (k_{\omega_i}^{01j})^{x_i} \cdot k_{\omega_i}^{00j}.
\end{aligned}$$

Тоді градієнт контролю (4.18) запишемо у вигляді

$$\gamma_i(x_i) = \mathbf{P}_Z \bullet \frac{\mathbf{P}_{A_i} \cdot \mathbf{B}_i^{x_i} \cdot (\mathbf{B}_i - \mathbf{I}) \cdot \mathbf{P}_{A_i}^{-1}(x_i)}{c_{\omega_i} + c_{U_i} \cdot \sum_{j=1, m} \left( p_{A_i}^{0j} \cdot (k_{\omega_i}^{01j})^{x_i} \cdot k_{\omega_i}^{00j} \right)} \cdot \mathbf{P}(\mathbf{X}), \quad (4.19)$$

де  $\mathbf{B}_i = \mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{U_i}$  – ймовірність переходу на доробку;

$\mathbf{P}_Z$  – матриця надійності оператора оновлення;

$\mathbf{I}$  – одинична матриця;

$\bullet$  – ліве домноження матриці.

В (4.19) ліве домноження матриці  $\mathbf{P}_Z$  забезпечує перетворення матриці на вектор, в якому перший елемент відповідає відношенню приросту безпомилковості до приросту вартості. Решта елементів – це відношення зменшення ймовірностей помилок різних типів до приросту вартості.

На розрахунки градієнтів за формулою (4.19) необхідно значно менше часу, ніж за формулою (4.18), тому що непотрібно для кожного нового варіанта АП перераховувати показники  $\mathbf{P}(\mathbf{X}(x_i = x_i + 1))$  та  $C(\mathbf{X}(x_i = x_i + 1))$ . В (4.19) показник  $\mathbf{P}(\mathbf{X})$  є однаковим для всіх  $x_i$ ,  $i = \overline{1, n}$  та обчислюється один раз, а в (4.18) для всіх  $x_i$ ,  $i = \overline{1, n}$  необхідно перераховувати показники  $\mathbf{P}(\mathbf{X}(x_i = x_i + 1))$  та  $C(\mathbf{X}(x_i = x_i + 1))$ .

#### 4.4.3. Градієнтний розподіл контрольних точок при обмеженнях на безпомилковість

Нижче пропонується градієнтний алгоритм оптимізації розподілу контрольних точок, яка забезпечує мінімальну вартість АП. Цей алгоритм складається з двох ітеративних ділянок. На першій ділянці послідовно встановлюються контрольні-добробчі оператори з максимальними градієнтами до отримання першого допустимого розв'язку, а на другій ділянці послідовно знімаються контролю для покращення розв'язку задачі (4.6).

Початковий вектор керованих змінних  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  будемо визначати з такої умови

$$x_i^{<0>} = \begin{cases} 1, & \text{якщо } p_{A_i}^1 < P^* \text{ або } p_{A_i}^{0j} > q_j, j = \overline{1, m}, \\ 0, & \text{інакше} \end{cases}, \quad i = \overline{1, n}. \quad (4.20)$$

Градієнтний алгоритм оптимізації розподілу контрольних точок (4.6) наведено нижче.

| *Перша ітеративна ділянка:*

⟨ Ініціалізувати  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  за формулою (4.20),  $N := 0$  ⟩;

⟨ Розрахувати  $p^1(\mathbf{X}^{<0>})$  та  $p^{0j}(\mathbf{X}^{<0>})$ ,  $j = \overline{1, m}$  за формулою (4.4) ⟩;

**While** ⟨  $p^1(\mathbf{X}^{<N>}) < P^*$  та  $p^{0j}(\mathbf{X}^{<N>}) > q_j$ ,  $j = \overline{1, m}$  ⟩

⟨ Ініціалізувати множину контролів  $L := \emptyset$  ⟩;

**For**  $i = 1 \dots n$  | для всіх робочих операторів

**If** ⟨  $x_i^{<N>} = 0$  ⟩

⟨ Обчислити градієнт контролю  $\gamma_i^{<N>}$  за формулою (4.19) ⟩;

⟨ Присвоїти  $L := \{L, i\}$  ⟩;

**End**

**End**

⟨ Визначити контроль з максимальним градієнтом

$l := \arg \max_{i \in L} \gamma_i^{<N>}$  ⟩;

⟨ Встановити  $x_l^{<N>} := 1$ ,  $N := N + 1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$  ⟩;

⟨ Обчислити  $p^1(\mathbf{X}^{<N>})$  та  $p^{0j}(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$  ⟩;

**End**

| *Друга ітеративна ділянка:*

**While** ⟨  $p^1(\mathbf{X}^{<N>}) \geq P^*$  та  $p^{0j}(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$  ⟩



```

⟨ Встановити  $N:=N+1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$  ⟩;
⟨ Ініціалізувати множину надлишкових контролів  $L := \emptyset$  ⟩;
For  $i=1\dots n$  | для всіх робочих операторів
  If ⟨  $x_i^{<N>} = 1$  ⟩;
    ⟨ Встановити  $x_i^{<N>} := 0$  ⟩;
    ⟨ Розрахувати  $p^1(\mathbf{X}^{<N>})$ ,  $p^{0_j}(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$  та  $C_i(\mathbf{X}^{<N>})$  ⟩;
    If ⟨  $p^1(\mathbf{X}^{<N>}) \geq P^*$  та  $p^{0_j}(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$  ⟩;
      ⟨ Присвоїти  $L := \{L, i\}$  ⟩;
    End
  ⟨ Встановити  $x_i^{<N>} := 1$  ⟩;
End
End
If ⟨  $L = \emptyset$  ⟩; Break | вихід з циклу While Else
  ⟨ Визначити контроль з мінімальною вартістю
   $l = \underset{i \in L}{\operatorname{arg\,min}} C_i(\mathbf{X}^{<N>})$  ⟩;
  ⟨ Встановити  $x_l^{<N>} := 0$  ⟩;
  ⟨ Обчислити  $p^1(\mathbf{X}^{<N>})$  та  $p^{0_j}(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$  ⟩;
End;
End.

```

В наведеному алгоритмі оптимізації вектори градієнтів  $\gamma_i^{<N>}$  ( $i = \overline{1, n}$ ) порівнюються за першою координатою, що відповідає відношенню приросту безпомилковості до приросту вартості.

Градієнтний алгоритм оптимізації розподілу контрольних точок протестовано на задачах А\_4\_1 та А\_4\_2. Результати оптимізації наведені в табл. 4.3.

На рис 4.2 порівнюється час оптимізації АП з швидким обчисленням градієнта контролю (4.19) та за формулою (4.18) при розв'язанні тестових задач А\_4\_1 та А\_4\_2. З рис. 4.2 видно, що на розрахунки градієнтів за формулою (4.19) необхідно значно менше часу, ніж за формулою (4.18).

На рис. 4.3 зображено протокол покрокової оптимізації розподілу контрольних точок для задачі А\_4\_2 з 20 робочими операторами ( $n=20$ ). Вісь, що ліворуч, відповідає ймовірності безпомилкового виконання АП, а вісь, що праворуч – ймовірностям помилок різних типів. Задача розподілу контрольних точок розв'язана за 11 ітерацій. На рис. 4.3 ітерації позначені числами від 1 до 11. З цього рисунка видно,

що на 9-й ітерації були виконанні обмеження по  $q_1 \dots q_4$ , а на 10-й ітерації – по  $p^1$ . На 11-й ітерації відбулося покращення розв’язку. В табл. 4.4 наведено протокол покрокової оптимізації задачі А\_4\_2.

Таблиця 4.3  
Результати градієнтної оптимізації РКТ тестових задач А 4 1 і А 4 2

Задача		$P^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	C
n=20	A 4 1	0,9635	0,0065	0,0087	0,0092	0,0122	226,50
	A 4 2	0,9575	0,0077	0,0093	0,0104	0,0151	157,29
n=40	A 4 1	0,9468	0,0094	0,0127	0,0134	0,0177	466,64
	A 4 2	0,9276	0,0127	0,0160	0,0182	0,0254	327,87
n=60	A 4 1	0,9310	0,0133	0,0165	0,0186	0,0207	554,76
	A 4 2	0,9015	0,0177	0,0210	0,0255	0,0343	488,11
n=80	A 4 1	0,8704	0,0235	0,0298	0,0335	0,0428	727,99
	A 4 2	0,8729	0,0245	0,0302	0,0343	0,0381	713,39
n=100	A 4 1	0,8566	0,0258	0,0322	0,0373	0,0481	1005,09
	A 4 2	0,8458	0,0292	0,0363	0,0412	0,0475	867,22
n=120	A 4 1	0,8397	0,0289	0,0367	0,0416	0,0531	1125,9
	A 4 2	0,8107	0,0357	0,0443	0,0503	0,0590	968,01

Таблиця 4.4  
Протокол покрокової оптимізації РКТ тестової задачі А 4 2 при n=20

№	X	C(X)	$p^1(X)$	$p^{0_1}(X)$	$p^{0_2}(X)$	$p^{0_3}(X)$	$p^{0_4}(X)$
0	00000000000000000000	100,90	0,9040	0,0173	0,0208	0,0243	0,0335
1	00010000000000000000	101,58	0,9096	0,0164	0,0193	0,0229	0,0317
2	00010000100000000000	102,83	0,9148	0,0154	0,0183	0,0216	0,0300
3	00010000100000000100	107,84	0,9207	0,0143	0,0169	0,0199	0,0282
4	00010000100000010100	114,51	0,9270	0,0133	0,0156	0,0183	0,0257
5	00010100100000010100	122,05	0,9339	0,0120	0,0141	0,0164	0,0235
6	00010100100010010100	124,79	0,9362	0,0116	0,0137	0,0159	0,0227
7	00010100101010010100	132,82	0,9428	0,0105	0,0125	0,0139	0,0204
8	00010100101010110100	141,74	0,9489	0,0093	0,0111	0,0123	0,0184
9	00010110101010110100	153,13	0,9560	0,0080	0,0096	0,0107	0,0156
10	00010111101010110100	160,54	0,9598	0,0072	0,0089	0,0098	0,0143

11	00010111101000110100	157,80	0,9575	0,0077	0,0093	0,0104	0,0151
----	----------------------	--------	--------	--------	--------	--------	--------

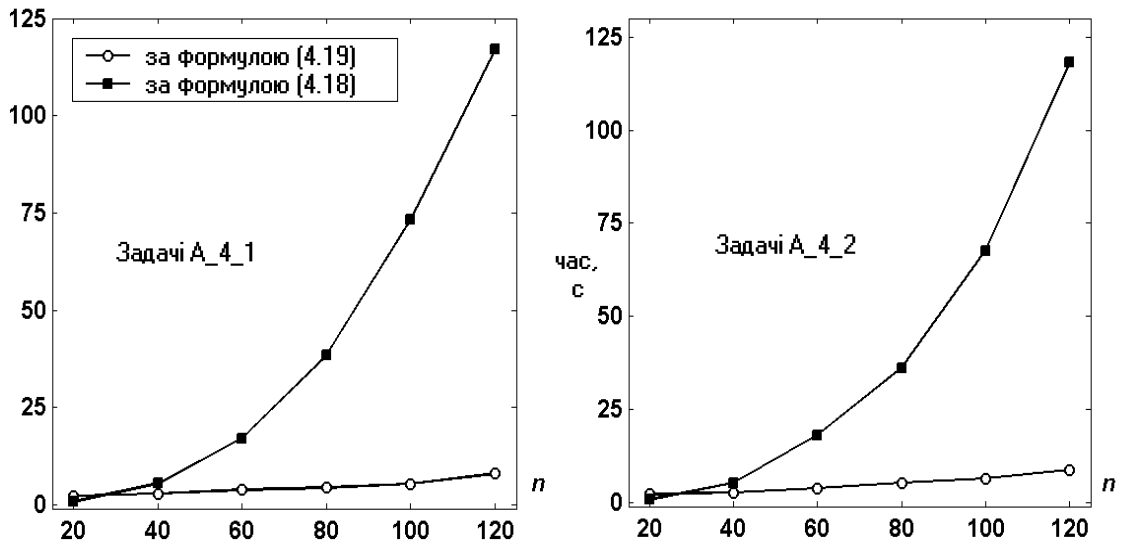


Рис. 4.2. Порівняння часу оптимізації АП з швидким обчислення градієнта контролю (4.19) та за формулою (4.18)

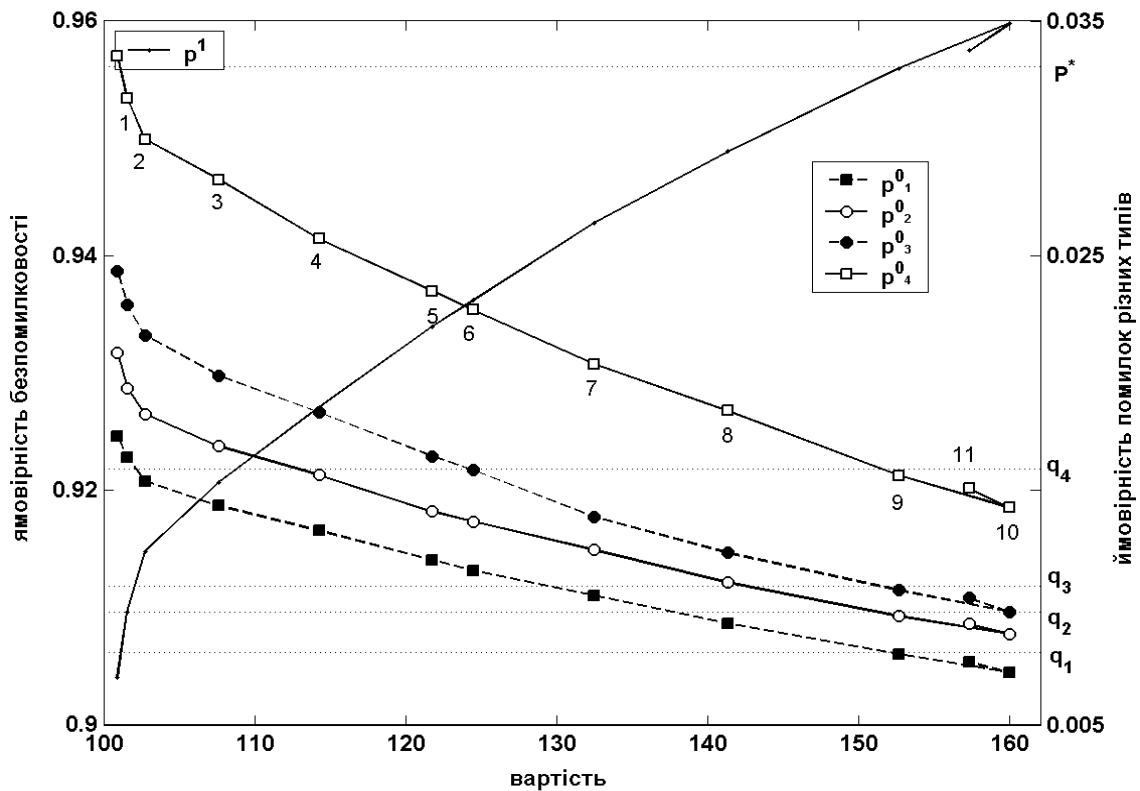


Рис. 4.3. Протокол покрокової оптимізації розподілу контрольних точок задачі A\_4\_1 при n=20

#### 4.4.4. Градієнтний розподіл контрольних точок при обмежених ресурсах

Нижче пропонується градієнтний алгоритм оптимізації розподілу контрольних точок при обмежених ресурсах. Цей алгоритм має дві ітеративні ділянки. На першій ділянці послідовно встановлюються контрольні-добробчі операції з максимальними градієнтами. При перевищенні допустимих ресурсів  $C^*$  починається друга ітеративна ділянка, на якій алгоритм оптимізації намагається повернути розв'язок задачі в допустиму область. Початковий вектор керованих змін  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  будемо визначати з такої умови

$$x_i^{<0>} = \begin{cases} 1, & \text{якщо } p_{A_i}^{0j} > q_j, j = \overline{1, m}, i = \overline{1, n}. \\ 0, & \text{інакше} \end{cases} \quad (4.21)$$

Градієнтний алгоритм оптимізації задачі розподілу контрольних точок наведено нижче.

| *Перша ітеративна ділянка:*

⟨ Ініціалізувати  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  за формулою (4.21) ⟩;

⟨ Розрахувати  $C(\mathbf{X}^{<0>})$  за формулою (4.5),  $N:=0$  ⟩;

**While** ⟨  $C(\mathbf{X}^{<N>}) < C^*$  ⟩

⟨ Ініціалізувати множину контролів  $L := \emptyset$  ⟩;

**For**  $i=1 \dots n$  | для всіх робочих операторів

**If** ⟨  $x_i^{<N>} = 0$  ⟩

⟨ Обчислити градієнт контролю  $\gamma_i^{<N>}$  за формулою (4.19) ⟩;

⟨ Присвоїти  $L := \{L, i\}$  ⟩;

**End**

**End**

⟨ Визначити контроль з максимальним градієнтом

$l := \arg \max_{i \in L} \gamma_i^{<N>}$  ⟩;

⟨ Встановити  $x_l^{<N>} := 1$ ,  $N:=N+1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$  ⟩;

⟨ Обчислити  $C(\mathbf{X}^{<N>})$  ⟩;

**End**

| *Друга ітеративна ділянка:*

**If** ⟨ Розв'язок  $\mathbf{X}$  не задовольняє умови  $p_j^0(\mathbf{X}^{<N>}) \leq q_j, j = \overline{1, m}$  ⟩

```

    < Задача немає розв'язків > Else
    < Зробити відкат на 1 крок назад:  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$  >;
    While <  $C(\mathbf{X}^{<N>}) \leq C^*$  >
        < Ініціалізувати множини додаткових контролів  $L := \emptyset$  >;
        For  $i=1 \dots n$  | для всіх робочих операторів
            If <  $x_i^{<N>} = 0$  >;
                < Встановити  $x_i^{<N>} := 1$  >;
                < Обчислити  $p_i^1(\mathbf{X}^{<N>})$ ,  $p^{0_j}(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$  та  $C(\mathbf{X}^{<N>})$  >;
                If <  $C(\mathbf{X}^{<N>}) \leq C^*$  та  $p^{0_j}(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$  >;
                    < Присвоїти  $L := \{L, i\}$  >;
                End
            End
            < Встановити  $x_i^{<N>} := 0$  >;
        End
    End
    If <  $L = \emptyset$  >; Break | вихід з циклу While Else
    < Визначити контроль з максимальною ймовірністю
     $l := \arg \max_{i \in L} p_i^1(\mathbf{X}^{<N>})$  >;
    < Встановити  $x_l^{<N>} := 1$ ,  $N := N+1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$  >;
    < Обчислити  $C(\mathbf{X}^{<N>})$  >;
    End;
End;
End.

```

Градiєнтний алгоритм оптимізації розподілу контрольних точок протестовано на задачах В\_4\_1 та В\_4\_2. Результати оптимізації наведені в табл. 4.5. Жирним шрифтом в табл. 4.5 виділені розв'язки, які не задовольняють обмеження з табл. 4.3.

На рис. 4.4 показано як знайдені розв'язки задовольняють обмеження (3.7) для тестових задач В\_4\_1 та В\_4\_2. На цих рисунках осям ординат відповідає такий відносний показник:

$$\text{для } C(\mathbf{X}): \psi(\mathbf{X}) = \frac{C^* - C(\mathbf{X})}{C^*} \cdot 100\%;$$

$$\text{для } p^{0_j}(\mathbf{X}): \psi(\mathbf{X}) = \frac{q_j - p^{0_j}(\mathbf{X})}{q_j} \cdot 100\%, \quad j = \overline{1, m}.$$

Від'ємне значення  $\psi(X)$  свідчить, що розв'язок  $X$  не задовольняє обмеження. Розв'язки, що не задовольняють обмеження, знайдені для тестових задач В\_4\_1 при  $n=20, 40, 100$  та  $120$ .

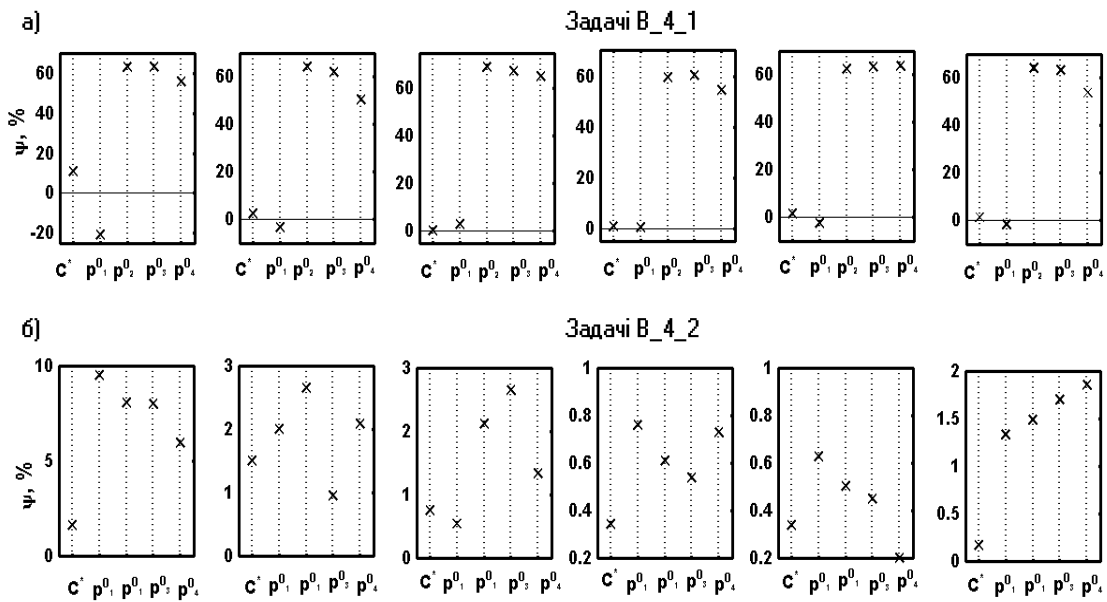


Рис. 4.4. Виконання оптимальними розв'язками обмежень (3.7)

Таблиця 4.5  
Результати градієнтної оптимізації РКТ тестових задач В\_4\_1 і В\_4\_2

задача		$C^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
$n=20$	В_4_1	201,45	<b>0,0067</b>	0,0088	0,0094	0,0125	0,9626
	В_4_2	176,94	0,0086	0,0114	0,0127	0,0143	0,9530
$n=40$	В_4_1	431,07	<b>0,0109</b>	0,0145	0,0156	0,0201	0,9389
	В_4_2	317,95	0,0148	0,0185	0,0208	0,0236	0,9224
$n=60$	В_4_1	539,49	0,0135	0,0167	0,0188	0,0206	0,9304
	В_4_2	521,86	0,0186	0,0232	0,0258	0,0296	0,9028
$n=80$	В_4_1	710,89	0,0241	0,0307	0,0342	0,0438	0,8672
	В_4_2	667,81	0,0262	0,0327	0,0370	0,0410	0,8631
$n=100$	В_4_1	963,90	<b>0,0268</b>	0,0337	0,0391	0,0511	0,8494
	В_4_2	826,34	0,0317	0,0396	0,0443	0,0498	0,8346
$n=120$	В_4_1	1126,38	<b>0,0283</b>	0,0360	0,0408	0,0519	0,8429
	В_4_2	1019,5	0,0370	0,0461	0,0519	0,0582	0,8068

На рис. 4.5 зображено протокол покрової оптимізації розподілу контрольних точок для задачі В\_4\_2 з 20 робочими операторами. Ця задача розв'язана за 11 ітерацій. На 9-й ітерації було виконано обмеження по  $q_1, q_2$  та  $q_3$ , а на 10-й ітерації – по  $q_4$  та  $p^1$  але порушено по  $C^*$ . На 11-й ітерації алгоритм оптимізації повернув розв'язок в допустиму область. В табл. 4.6 наведено протокол покрової оптимізації цієї задачі.

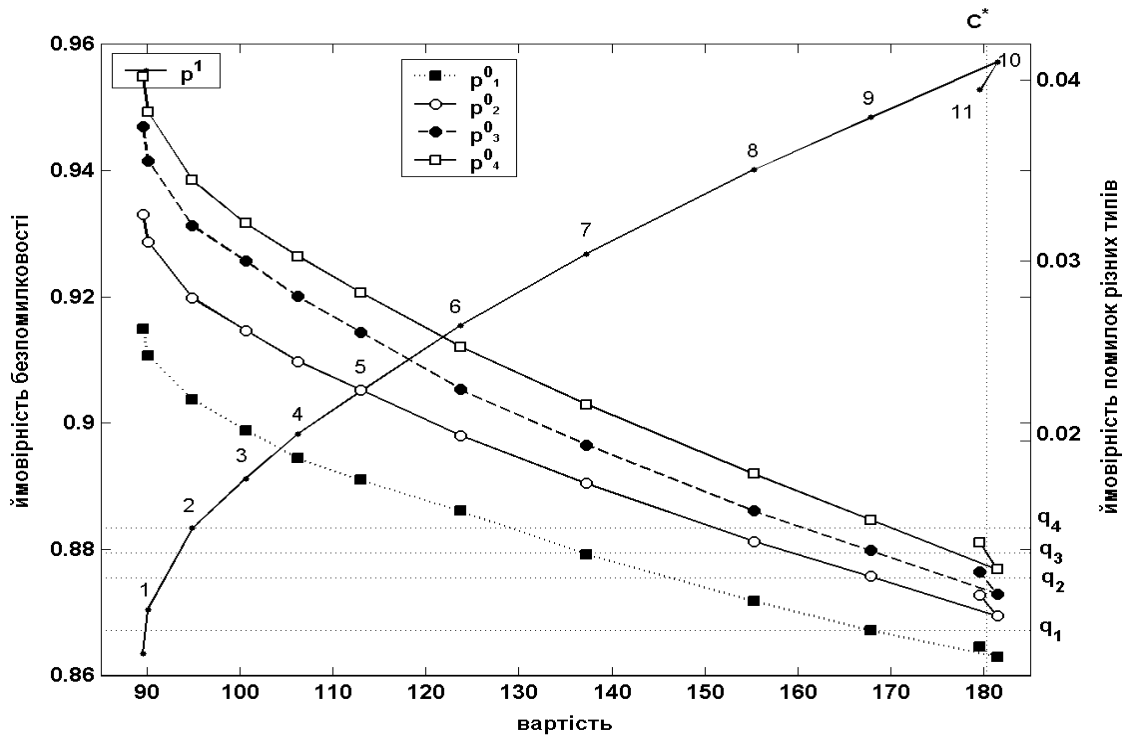


Рис. 4.5. Протокол покрової оптимізації розподілу контрольних точок задачі В\_4\_2 при n=20

Таблиця 4.6  
Протокол покрової оптимізації РКТ тестової задачі В\_4\_1 при n=20

N	X	C(X)	$p^1(X)$	$p^{0_1}(X)$	$p^{0_2}(X)$	$p^{0_3}(X)$	$p^{0_4}(X)$
0	00000000000000000000	89,52	0,8635	0,0262	0,0326	0,0375	0,0402
1	0000000000000000000100	90,17	0,8705	0,0247	0,0310	0,0355	0,0382
2	0000001000000000000100	95,00	0,8834	0,0223	0,0279	0,0319	0,0345
3	0000011000000000000100	100,76	0,8912	0,0206	0,0261	0,0300	0,0321
4	0000111000000000000100	106,38	0,8983	0,0190	0,0244	0,0280	0,0302
5	000011100000000001100	113,16	0,9050	0,0179	0,0228	0,0260	0,0282
6	000011100000000101100	124,01	0,9155	0,0161	0,0203	0,0229	0,0252

7	00101111000000101100	137,61	0,9268	0,0137	0,0177	0,0198	0,0220
8	00101111000000101100	155,65	0,9401	0,0111	0,0144	0,0161	0,0182
9	00101111100000101100	168,21	0,9484	0,0095	0,0125	0,0139	0,0156
10	00101111100000101101	181,93	0,9573	0,0080	0,0103	0,0115	0,0129
11	00101111100000101110	179,23	0,9524	0,0088	0,0115	0,0128	0,0145

#### 4.4.5. Градієнтний вибір кратностей контролів при обмеженнях на безпомилковість

Нижче пропонується градієнтний алгоритм оптимізації вибору кратностей контролів, яка забезпечує мінімальну вартість АП. Він складається з двох ітеративних ділянок. На першій ділянці відбувається поелементне нарощування кратностей контролів до отримання першого допустимого розв'язку, а на другій ділянці відбувається покращення розв'язку за рахунок зняття контролів з максимальною вартістю, для яких не порушуються обмеження задачі (4.19).

Градієнтний алгоритм оптимізації задачі вибору кратностей контролів наведено нижче.

| *Перша ітеративна ділянка:*

⟨Ініціалізувати  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  за формулою (4.20),  $N:=0$ ⟩;

⟨Розрахувати  $p^1(\mathbf{X}^{<0>})$  та  $p^{0_j}(\mathbf{X}^{<0>})$ ,  $j = \overline{1, m}$  за формулою (4.13)⟩;

**While** ⟨  $p^1(\mathbf{X}^{<N>}) < P^*$  та  $p^{0_j}(\mathbf{X}^{<N>}) > q_j$ ,  $j = \overline{1, m}$  ⟩

**For**  $i=1 \dots n$  | для всіх робочих операторів

⟨Обчислити градієнт контролю  $\gamma_i^{<N>}$  за формулою (4.19)⟩;

**End**

⟨Визначити контроль з максимальним градієнтом

$l = \arg \max_{i \in L} \gamma_i^{<N>}$ ⟩;

⟨Встановити  $x_l^{<N>} := x_l^{<N>} + 1$ ,  $N:=N+1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$ ⟩;

⟨Обчислити  $p^1(\mathbf{X}^{<N>})$  та  $p^{0_j}(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$ ⟩;

**End**

| *Друга ітеративна ділянка:*

**While** ⟨  $p^1(\mathbf{X}^{<N>}) \geq P^*$  та  $p^{0_j}(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$  ⟩

⟨Встановити  $N:=N+1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$ ⟩;

⟨Ініціалізувати множину надлишкових контролів  $L := \emptyset$ ⟩;

**For**  $i=1 \dots n$  | для всіх робочих операторів

⟨Встановити  $x_i^{<N>} := x_i^{<N>} - 1$ ⟩;



$\langle \text{Розрахувати } p^1(\mathbf{X}^{<N>}), p^{0_j}(\mathbf{X}^{<N>}), j = \overline{1, m} \text{ та } C_i(\mathbf{X}^{<N>}) \rangle;$   
**If**  $\langle p^1(\mathbf{X}^{<N>}) \geq P^* \text{ та } p^{0_j}(\mathbf{X}^{<N>}) \leq q_j, j = \overline{1, m} \rangle;$   
      $\langle \text{Присвоїти } L := \{L, i\} \rangle;$   
**End**  
      $\langle \text{Встановити } x_i^{<N>} := x_i^{<N>} + 1 \rangle;$   
**End**  
**If**  $\langle L = \emptyset \rangle;$  **Break** | вихід з циклу *While*  
**Else**  
      $\langle \text{Визначити контроль з мінімальною вартістю} \rangle$   
      $l = \underset{i \in L}{\operatorname{arg\,min}} C_i(\mathbf{X}^{<N>});$   
      $\langle \text{Встановити } x_l^{<N>} := x_l^{<N>} - 1 \rangle;$   
      $\langle \text{Обчислити } p^1(\mathbf{X}^{<N>}) \text{ та } p^{0_j}(\mathbf{X}^{<N>}), j = \overline{1, m} \rangle;$   
**End;**  
**End.**

Алгоритм оптимізації вибору кратностей контролів протестовано на задачах А\_4\_1 та А\_4\_2. Результати оптимізації наведені в табл. 4.7. Порівняння табл. 4.3 та табл. 4.7 показує, що перехід від задачі розподілення контрольних точок до задачі вибору кратностей контролів дозволяє при тих самих ресурсах синтезувати АП з кращими показниками надійності, особливо для задач А\_4\_2.

Таблиця 4.7

Результати градієнтної оптимізації ВКК тестових задач А\_4\_1 і А\_4\_2

задача		$p^1$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$C$
$n = 20$	А_4_1	0,9632	0,0066	0,0087	0,0093	0,0123	203,02
	А_4_2	0,9577	0,0077	0,0093	0,0103	0,0150	154,37
$n = 40$	А_4_1	0,9471	0,0093	0,0126	0,0133	0,0178	460,99
	А_4_2	0,9271	0,0128	0,0162	0,0183	0,0256	325,25
$n = 60$	А_4_1	0,9307	0,0134	0,0166	0,0187	0,0206	540,25
	А_4_2	0,9011	0,0177	0,0212	0,0257	0,0342	482,24
$n = 80$	А_4_1	0,8703	0,0236	0,0298	0,0333	0,0429	719,90
	А_4_2	0,8729	0,0244	0,0302	0,0344	0,0381	700,97
$n = 100$	А_4_1	0,8548	0,0257	0,0326	0,0378	0,0491	992,01
	А_4_2	0,8452	0,0294	0,0364	0,0413	0,0476	860,59
$n = 120$	А_4_1	0,8410	0,0287	0,0364	0,0412	0,0527	1118,05
	А_4_2	0,8112	0,0356	0,0442	0,0502	0,0588	962,69

На рис 4.6 порівнюються час оптимізації АП з швидким обчисленням градієнта контролю (4.19) та за формулою (4.18) для задач А\_4\_1 та А\_4\_2.

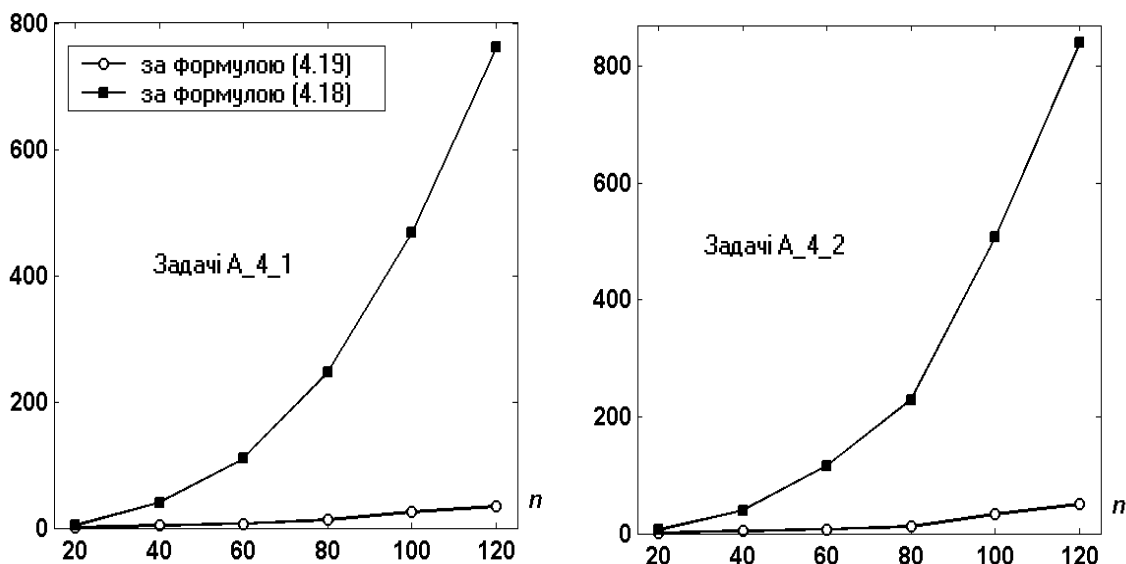


Рис. 4.6. Порівняння часу оптимізації АП з швидким обчисленням градієнта контролю (4.19) та за формулою (4.18)

#### 4.4.6. Градієнтний вибір кратностей контролів при обмежених ресурсах

Нижче пропонується градієнтний алгоритм оптимізації вибору кратностей контролів при обмежених ресурсах. Цей алгоритм складається з двох ітеративних ділянок. На першій ділянці аналогічно попередньому алгоритму оптимізації відбувається поелементне нарощування контролів до отримання першого допустимого розв'язку, а на другій – покращення розв'язку за рахунок розподілення ресурсів, що залишилися.

Градієнтний алгоритм оптимізації задачі вибору кратностей контролів наведено нижче.

| *Перша ітеративна ділянка:*

⟨ Ініціалізувати  $\mathbf{X}^{<0>} = (x_1^{<0>}, x_2^{<0>}, \dots, x_n^{<0>})$  за формулою (4.21) ⟩;

⟨ Розрахувати  $C(\mathbf{X}^{<0>})$  за формулою (4.14),  $N:=0$  ⟩;

**While** ⟨  $C(\mathbf{X}^{<N>}) < C^*$  ⟩

**For**  $i=1\dots n$  | для всіх робочих операторів  
 ‹Обчислити градієнт контролю  $\gamma_i^{<N>}$  за формулою (4.19)›;  
**End**  
 ‹Визначити контроль з максимальним градієнтом  
 $l := \arg \max_{i \in L} \gamma_i^{<N>}$ ›;  
 ‹Встановити  $x_l^{<N>} := x_l^{<N>} + 1$ ,  $N := N + 1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$ ›;  
 ‹Обчислити  $C(\mathbf{X}^{<N>})$ ›;  
**End**  
 | Друга ітеративна ділянка:  
**If** ‹Розв'язок  $\mathbf{X}$  не задовольняє умови  $p_j^0(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$ ›  
 ‹Задача немає розв'язків›;  
**Else**  
 ‹Зробити відкат на 1 крок назад  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$ ›;  
**While** ‹ $C(\mathbf{X}^{<N>}) \leq C^*$ ›  
 ‹Ініціалізувати множини додаткових контролів  $L := \emptyset$ ›;  
**For**  $i=1\dots n$  | для всіх робочих операторів  
 ‹Встановити  $x_i^{<N>} := x_i^{<N>} - 1$ ›;  
 ‹Обчислити  $p_i^1(\mathbf{X}^{<N>})$ ,  $p_j^0(\mathbf{X}^{<N>})$ ,  $j = \overline{1, m}$  та  $C(\mathbf{X}^{<N>})$ ›;  
**If** ‹ $C(\mathbf{X}^{<N>}) \leq C^*$  та  $p_j^0(\mathbf{X}^{<N>}) \leq q_j$ ,  $j = \overline{1, m}$ ›;  
 ‹Присвоїти  $L := \{L, i\}$ ›;  
**End**  
 ‹Встановити  $x_i^{<N>} := x_i^{<N>} - 1$ ›;  
**End**  
**If** ‹ $L = \emptyset$ ›; **Break** | вихід з циклу *While*  
**Else**  
 ‹Визначити контроль з максимальною ймовірністю  
 $l := \arg \max_{i \in L} p_i^1(\mathbf{X}^{<N>})$ ›;  
 ‹Встановити  $x_l^{<N>} := x_l^{<N>} + 1$ ,  $N := N + 1$ ,  $\mathbf{X}^{<N>} := \mathbf{X}^{<N-1>}$ ›;  
 ‹Обчислити  $C(\mathbf{X}^{<N>})$ ›;  
**End**;  
**End**;  
**End**.

Градiєнтний алгоритм оптимiзацiї вибору кратностей контролiв протестовано на задачах В\_4\_1 та В\_4\_2. Результати оптимiзацiї наведенi в табл. 4.8. Жирним шрифтом в табл. 4.8 видiленi розв'язки, якi не задовольняють обмеження з табл. 4.2. Розв'язки, що не задовольняють обмеження, знайденi для тестових задач В\_4\_2 при  $n=20$ ,  $n=60$  та  $n=100$ .

Таблиця 4.8  
Результати градiєнтної оптимiзацiї ВКК тестових задач В\_4\_1 i В\_4\_2

Задача		$C$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
$n=20$	В_4_1	205,15	<b>0,0063</b>	0,0082	0,0088	0,0115	0,9652
	В_4_2	179,12	0,0087	0,0110	0,0123	0,0140	0,9539
$n=40$	В_4_1	441,54	0,0103	0,0137	0,0149	0,0188	0,9423
	В_4_2	313,21	0,0147	0,0184	0,0209	0,0235	0,9225
$n=60$	В_4_1	522,44	<b>0,0143</b>	0,0179	0,0202	0,0222	0,9254
	В_4_2	525,73	0,0184	0,0230	0,0255	0,0293	0,9037
$n=80$	В_4_1	714,96	0,0239	0,0300	0,0341	0,0431	0,8689
	В_4_2	670,10	0,0260	0,0324	0,0366	0,0408	0,8642
$n=100$	В_4_1	969,18	<b>0,0263</b>	0,0330	0,0384	0,0499	0,8524
	В_4_2	829,10	0,0315	0,0391	0,0438	0,0494	0,8363
$n=120$	В_4_1	1140,91	0,0275	0,0348	0,0397	0,0501	0,8479
	В_4_2	1021,16	0,0367	0,0459	0,0517	0,0578	0,8080

На рис. 4.7 порiвнюються розв'язки тестових задач В\_4\_1 та В\_4\_2, що задовольняють обмеження з табл. 4.2 при розв'язанi задач розподiлу контрольних точок та вибору кратностей контролiв.

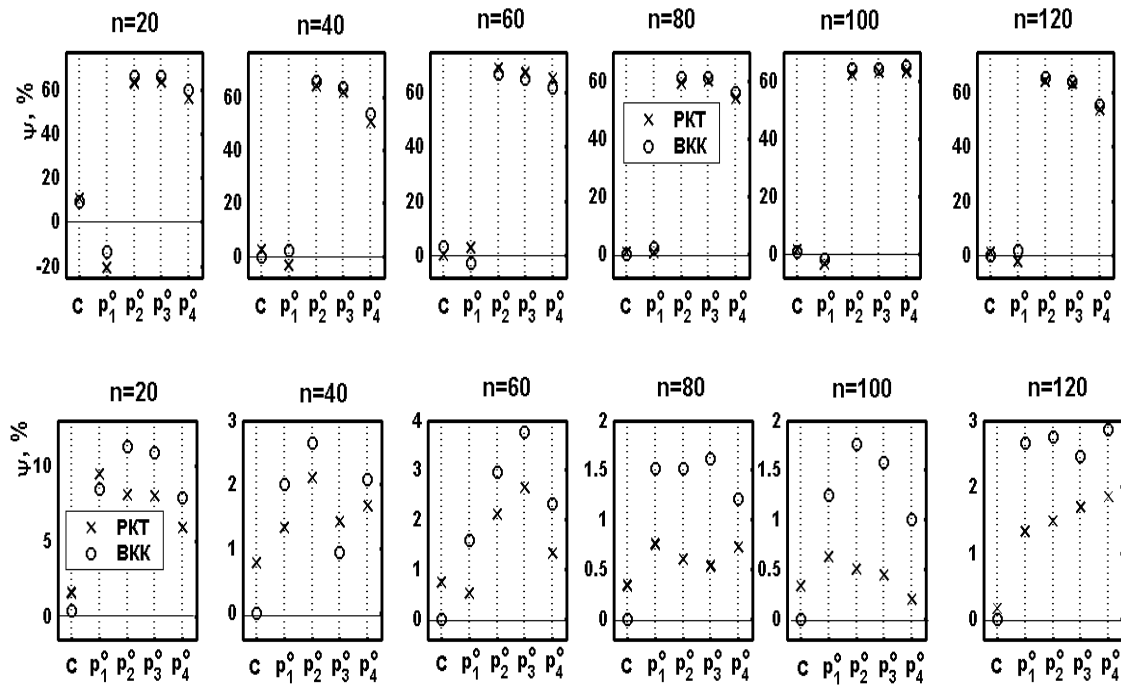


Рис. 4.7 Виконання оптимальними розв'язками обмежень (3.7)

З рис. 4.7 видно, що градієнтні алгоритми оптимізації розподілу контрольних точок та вибору кратностей контролів при обмежених ресурсах добре працюють, коли допустимі ймовірності наявності помилок різних типів приблизно однакові. Для задач, в яких ці ймовірності відрізняються суттєво, запропоновані алгоритми не завжди знаходять допустимий розв'язок при встановлених ресурсних обмеженнях. Для покращення розв'язку в градієнтних алгоритмах необхідно в другій ітеративній ділянці збільшувати кількість кроків відкату назад. З ростом кількості таких кроків складність алгоритму і відповідно витрати часу на пошук оптимуму зростають за експоненційним законом. Тому, виникає необхідність в розробці алгоритмів оптимізації, які б за прийнятний час знаходили розв'язок близький до оптимального, а при відсутності дефіциту часу знаходили глобальний екстремум. Наступних два розділа присвячено розробці таких алгоритмів, які ґрунтуються на генетичних принципах оптимізації.

## 5. ГЕНЕТИЧНА ОПТИМІЗАЦІЯ НАДІЙНОСТІ БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

Пропонуються ГА оптимізації надійності багатовимірних алгоритмічних процесів. Проводиться порівняльний аналіз запропонованих градієнтних та ГА оптимізацій за теоретичною складністю, практичною швидкістю та якістю знайденого розв'язку. В розділі використовуються матеріали робіт [72, 86, 91, 109, 110, 133, 139, 152, 174, 175].

### 5.1. Ідея генетичного пошуку

ГА оптимізації представляє собою стохастичний пошук, що базується на механізмах еволюції живої природи [107, 139, 154, 156]. Основу ГА оптимізації складають поняття хромосома, ген, популяція та фітнес-функція. В класичній теорії оптимізації цим поняттям відповідають вектор керованих змінних, керована змінна, множина розв'язків та цільова функція.

Принципова відмінність ГА оптимізацій від класичних методів оптимізації полягає в тому, що вони не використовують частинні похідні при виборі напрямку пошуку оптимуму, а базуються на операціях схрещування, мутації і селекції.

Схрещування представляє собою операцію  $C$ , що виконується над двома батьківськими хромосомами, в результаті якої отримуються дві хромосоми-нащадки, що успадковують частину генів від кожної батьківської хромосоми.

Мутація хромосоми представляє собою операцію  $M$ , в якій відбувається випадкова зміна значення гена.

Селекція  $S$  – це вибір у нову популяцію найкращих хромосом відповідно до значень фітнес-функції.

Пошук оптимуму в ГА оптимізації відбувається одночасно з множини початкових точок  $\{X_1, X_2, \dots, X_{pop\_size}\}$ , які складають популяцію з  $pop\_size$  хромосом. Оптимізація відбувається шляхом багатократного виконання операцій схрещування  $C$ , мутації  $M$  та селекції  $S$ . Типовий ГА оптимізації реалізується такою процедурою [72, 139]:

*⟨Ініціалізувати параметри генетичного алгоритму:*

*pop\_size – розмір популяції; num\_iter – кількість ітерацій; num\_cross – кількість операцій схрещувань; num\_mut – кількість операцій мутацій ⟩;*

⟨Ініціалізувати популяцію  $\mathbf{X}^{<0>} = \{\mathbf{X}_1^{<0>}, \mathbf{X}_2^{<0>}, \dots, \mathbf{X}_{pop\_size}^{<0>}\}$ ⟩;  
 ⟨Оцінити фітнес-функцію хромосом початкової популяції  $\mathbf{X}^{<0>}$ ⟩;  
**For**  $t=1 \dots num\_iter$   
   ⟨Ініціалізувати  $\mathbf{X}^{<C>} := \emptyset$ ⟩;  
   **For**  $i=1 \dots num\_cross$   
     ⟨Вибрати з популяції  $\mathbf{X}^{<t>}$  дві хромосоми  $\mathbf{X}_{k1}^{<t>}$  та  $\mathbf{X}_{k2}^{<t>}$ ⟩;  
     ⟨Виконати операцію схрещування  $(\mathbf{X}_1, \mathbf{X}_2) = C(\mathbf{X}_{k1}^{<t>}, \mathbf{X}_{k2}^{<t>})$ ⟩;  
     ⟨Присвоїти  $\mathbf{X}^{<C>} := \{\mathbf{X}^{<C>}, \mathbf{X}_1, \mathbf{X}_2\}$ ⟩;  
   **End;**  
   ⟨Ініціалізувати  $\mathbf{X}^{<M>} := \emptyset$ ⟩;  
   **For**  $i=1 \dots num\_mut$   
     ⟨Вибрати з популяції  $\mathbf{X}^{<t>}$  хромосому  $\mathbf{X}_k^{<t>}$ ⟩;  
     ⟨Виконати операцію мутації  $\mathbf{X} = M(\mathbf{X}_k^{<t>})$ ⟩;  
     ⟨Присвоїти  $\mathbf{X}^{<M>} := \{\mathbf{X}^{<M>}, \mathbf{X}\}$ ⟩;  
   **End;**  
   ⟨Оцінити фітнес-функцію нових хромосом  $\mathbf{X}^{<C>}$  та  $\mathbf{X}^{<M>}$ ⟩;  
   ⟨Виконати селекцію  $\mathbf{X}^{<t+1>} = S(\{\mathbf{X}^{<t>}, \mathbf{X}^{<C>}, \mathbf{X}^{<M>}\})$ ⟩;  
**End.**

При застосуванні ГА для розв'язання конкретних задач оптимізації потрібно реалізувати операції кодування, схрещування, мутації та селекції. Нижче пропонуються реалізації цих генетичних операцій.

## 5.2. Генетичне кодування варіанта алгоритмічного процесу

В підрозділі пропонуються схеми генетичного кодування варіантів АП для задач структурної, параметричної, та структурно-параметричної оптимізації. Розглядаються як часткові, так і загальна задачі структурної оптимізації АП.

### 5.2.1. Задача оптимізації контролів

Розглядаються задачі розстановки контрольних точок та вибору кратностей контролів. Ці задачі структурної оптимізації АП мають номери 1 та 2 за класифікацією з табл. 3.1. Для цих задач  $\mathbf{X} = \mathbf{X}_c$ , причому кількість керованих змінних дорівнює  $n$ . Тому варіант АП закодуємо хромосомою з  $n$  генів:  $\mathbf{X} = (x_{c1}, x_{c2}, \dots, x_{cn})$ , які набувають дискретних значень. Для задачі розподілу контрольних точок  $i$ -ий ген вказує на наявність ( $x_{ci} = 1$ ) або відсутність ( $x_{ci} = 0$ ) контролю  $\omega_i$  з

добробкою  $U_i$  (або з оновленням  $Z_i$ ) після  $i$ -го робочого оператора  $A_i$ ,  $i = \overline{1, n}$ . При виборі кратностей контролів  $x_{ci} \in \{0, 1, 2, \dots, \overline{x_{ci}}\}$  задає кількість таких покращувальних процедур, де  $\overline{x_{ci}}$  – апріорі визначена верхня межа кратності  $i$ -го контролю. Для алгоритмічних структур з пам'яттю кратність контролю інтерпретується як максимально допустима кількість перевірок.

На рис. 5.1 наведено приклад генетичного кодування для задачі розподілу контрольних точок в АП з 10-ти потенційно-контрольованих робочих операторів. У цьому прикладі контрольні точки встановлені після 1-го, 5-го, 7-го і 10-го робочих операторів.

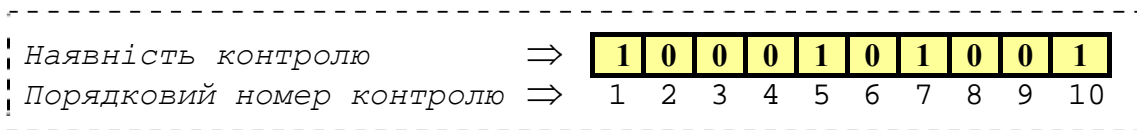


Рис. 5.1. Приклад кодування хромосоми для задачі розстановки контролів

На рис. 5.2 наведено приклад генетичного кодування для задачі оптимізації кратностей контролів. Хромосома з цього рисунку вказує, що після 1-го, 5-го, 7-го та 10-го робочих операторів встановлені однократні контролі, а після 2-го та 4-го робочих операторів – двократні контролі.

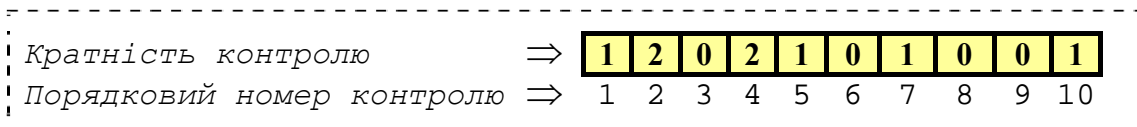


Рис. 5.2. Приклад кодування хромосоми для задачі оптимізації кратностей контролів

### 5.2.2. Задача вибору варіантів реалізацій операторів та логічних умов

Розглядається задача структурної оптимізації АП про вибір реалізацій операторів та логічних умов. Відповідно до підрозділу 3.1 для цієї задачі оптимізації  $\mathbf{X} = \mathbf{X}_a$ , причому кількість керованих змінних дорівнює  $h$ . Тому варіант АП закодуємо хромосомою з  $h$  генів:  $\mathbf{X} = (x_{a1}, x_{a2}, \dots, x_{ah})$ . Значення гена  $x_{ai} \in \{1, 2, \dots, \overline{x_{ai}}\}$  вказує на порядковий номер способу реалізації  $i$ -го елемента АП (оператора або логічної умови), де  $\overline{x_{ai}}$  – кількість можливих варіантів  $i$ -го елемента. Таким чином, гени хромосоми набувають дискретних значень. На рис. 5.3 наведено приклад генетичного кодування для АП, в якому 1-ий, 9-ий та 10-ий елементи АП реалізовані 1-им способом, 2-ий,



3-ій, 4-ий та 7-ий елементи реалізовані 2-им способом, а 5-ий, 6-ий та 8-ий елементи – 3-м способом.

Номер способу реалізації $\Rightarrow$	1	2	2	2	3	3	2	3	1	1
Порядковий номер елемента $\Rightarrow$	1	2	3	4	5	6	7	8	9	10

Рис. 5.3. Приклад кодування хромосоми для задачі вибору варіантів реалізацій операторів та логічних умов

### 5.2.3. Загальні задачі структурної оптимізації алгоритмічних процесів

В загальних задачах структурної оптимізації одночасно обирають способи реалізацій елементів АП та розставляють контрольні точки (або визначають кратності контролів). Ці задачі оптимізації мають номери 5 та 6 за класифікацією з табл. 3.1. Для них вектор керованих змінних визначається таким чином:  $\mathbf{X} = \mathbf{X}_a \cup \mathbf{X}_c$ . При цьому можливі дві ситуації: 1) коли контролі, для яких визначаються оптимальні кратності, реалізовані лише одним способом та 2) коли необхідно не тільки обрати найкращий спосіб реалізації деяких контролів, але і визначити чи потрібно на цьому місці ставити контроль та якою має бути його кратність. Розглянемо генетичне кодування варіанта АП для кожної ситуації.

Для *першої ситуації*  $\mathbf{X}_a \cap \mathbf{X}_c = \emptyset$ , тому варіант АП генетично представимо простим поєднанням хромосом з попередніх задач (див. пункти 5.2.1 та 5.2.2). Загальна кількість генів в цій хромосомі становить  $h + n$ .

Для *другої ситуації*  $\mathbf{X}_a \cap \mathbf{X}_c \neq \emptyset$ , тому варіант АП закодуємо хромосомою з  $(h + n - d)$  генів, де  $d = |\mathbf{X}_a \cap \mathbf{X}_c|$  – кількість контролів, які відображені і в  $\mathbf{X}_a$ , і в  $\mathbf{X}_c$ . Запропонована хромосома складається з хромосоми задачі вибору способів реалізації елементів АП та тих генів  $x_c$  хромосоми задачі оптимізації контролів для яких  $x_c \notin \mathbf{X}_a$ . Керовані змінні, що описують контролі лише з однієї з двох множин  $\mathbf{X}_a$  або  $\mathbf{X}_c$ , кодуємо як в попередніх пунктах. Кодування генами керованих змінних з множини  $\mathbf{X}_a \cap \mathbf{X}_c$  розглянемо для задач №4 та №5 окремо.

В задачі №4 одночасно розставляються контрольні точки та обираються способи реалізацій елементів АП (див. табл. 3.1). Позначимо через  $x_j$  ті контролі, які відображені і в  $\mathbf{X}_a$ , і в  $\mathbf{X}_c$ ,  $j = \overline{1, d}$ . Тоді, відповідний ген в хромосомі набуватиме значень  $x_j \in \{0, 1, 2, \dots, \overline{x_j}\}$ , які

вказують на порядковий номер способу реалізації контролю;  $\bar{x}_j$  – кількість способів реалізації контролю  $x_j$ ; нульове значення відповідає відсутності контролю.

В задачі №5 одночасно визначаються кратності контролів та обираються способи реалізацій елементів АП (див. табл. 3.1). Позначимо через  $m_j$  – верхню межу кратності контролю  $x_j$ . Тоді ген, що відповідає за цей контроль, прийматиме одне з  $(m_j \bar{x}_j + 1)$  можливих значень відповідно до рис. 5.4.

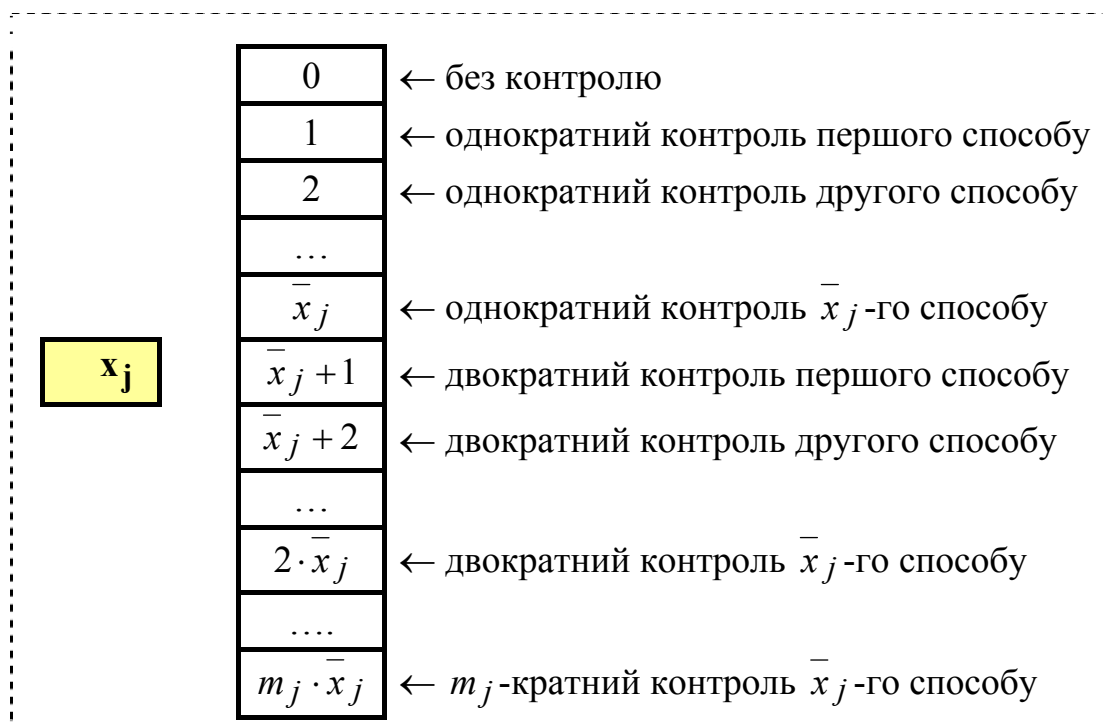


Рис. 5.4. Генетичне кодування контролю  $x_j$  за комбінаторною схемою для загальної задачі структурної оптимізація АП

Вище описана комбінаторна схема кодування є компактною, але дещо складною, особливо коли відрізняються максимальні кратності різних реалізацій контролів. Нижче пропонується проста альтернативна схема кодування контролю за допомогою декількох генів. Кількість генів відповідає числу реалізацій контролю. Кожен з генів задає кратність відповідної реалізації операції. Наприклад, хромосомою з рис. 5.5 закодована така інформація: 1-ий контроль реалізовано 3-м способом; 2-ий контроль відсутній; 3-ій контроль реалізовано 2-м способом з кратністю 3. Вводячи обмеження на значення генів за цією схемою можна закодувати різні варіанти управління надійністю. Наприклад, якщо кратні контрольні операції є однаковими, тоді серед усіх генів лише один може мати ненульове значення. Якщо дозволені

перевірки різними способами, тоді це обмеження зайве. Якщо обмежена максимальна кратність перевірок, тоді сума значень генів не повинна перевищувати відповідне порогове значення.

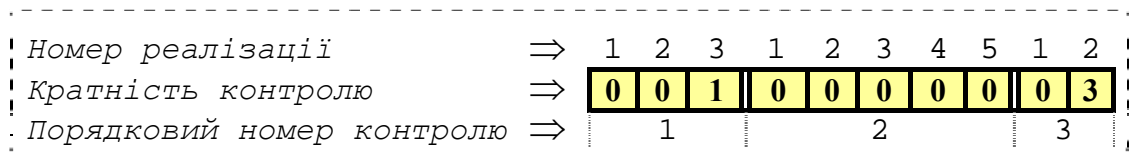


Рис. 5.5. Приклад кодування контролів за розширеною схемою для загальної задачі структурної оптимізація АП

#### 5.2.4. Задачі параметричної оптимізації алгоритмічних процесів

В задачах параметричної оптимізації обираємо значення факторів, що впливають на характеристики надійності операторів та логічних умов АП. Для цих задач оптимізації  $\mathbf{X} = \mathbf{X}_f$ , причому кількість керованих змінних дорівнює  $k$ . Ці задачі в табл. 3.1 записані під №3.

Позначимо через  $F$  – кількість елементів АП, надійнісні характеристики яких можна змінювати шляхом варіювання факторами. Фактори впливу на характеристики надійності  $i$ -го елемента позначимо через  $(f_1^i, f_2^i, \dots, f_{V_i}^i)$ , причому  $\sum_{i=1, F} V_i = k$ . Тоді варіант АП пропону-

ється закодувати хромосомою з рис. 5.6. Гени цієї хромосоми можуть набувати як дискретних, так і неперервних значень, відповідно до можливостей варіювання факторами.



Рис. 5.6. Кодування хромосоми для задачі параметричної оптимізація АП

#### 5.2.5. Задачі структурно-параметричної оптимізації алгоритмічних процесів

В задачах структурно-параметричної оптимізації одночасно встановлюємо значення факторів, що впливають на характеристики надійності операторів та логічних умов, та обираємо структуру АП – визначаємо способи реалізацій елементів АП, розставляємо контрольні точки або визначаємо їх кратності. Ці задачі оптимізації в табл. 3.1 мають номери з 7 по 11. Для них вектор керованих змінних визначається та-

ким чином:  $X = X_a \cup X_f$ , або  $X = X_c \cup X_f$ , або  $X = X_a \cup X_c \cup X_f$ .

При цьому можливі дві ситуації: 1) коли керовані змінні з структурної та параметричної оптимізації стосуються різних елементів АП та 2) коли деякі елементи АП беруть участь і в структурній, і в параметричній оптимізації. Розглянемо генетичне кодування варіанта АП для кожної ситуації.

В *першій ситуації* керовані змінні незалежні, тому варіант АП генетично представимо простим поєднанням хромосом з задач структурної оптимізації та з параметричної оптимізації (див. пункти 5.2.4). Графічне зображення цієї хромосоми наведено на рис. 5.7.

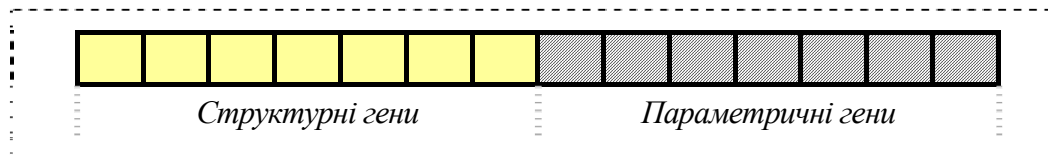


Рис. 5.7. Кодування хромосоми для задачі структурно-параметричної оптимізації АП при незалежних керованих змінних

В *другій ситуації* деякі керовані змінні є залежними. Для прикладу розглянемо випадок, коли відомими є декілька способів реалізації певного робочого оператора. Причому для деяких варіантів цього оператора відомі залежності характеристик надійності від факторів впливу. Перелік факторів впливу для різних реалізацій оператора може бути різним. Для врахування можливості варіювання не тільки реалізаціями оператора, але і факторами впливу, пропонується хромосома з вкладеною структурою генів (рис. 5.8). Вона складається з генів трьох типів: структурних, параметричних та змішаних. Структурні та параметричні гени такі самі, як і в хромосомі з рис. 5.7. Змішані гени задають структурні елементи АП одночасно зі значеннями контрольованих факторів, які впливають на їх надійність. В хромосомі з рис. 5.8 перші 10 генів задають структуру АП. Змішані гени з номерами 3 та 8 задають ще і значення факторів впливу для відповідних структурних елементів. На характеристики надійності 3-го структурного елемента впливає 4 фактори, а на характеристики 8-го елемента – 3 фактори. Зазначимо, що якщо змішаний ген задає спосіб реалізації елемента АП, тоді кількість його параметричних генів може бути різною і визначатися значенням структурного гена. В цьому випадку доцільно використовувати розширене кодування з рис. 5.5. У випадку малої кількості комбінацій значень факторів впливу, кожну таку комбінацію можна представити як окрему реалізацію елемента АП. Це дозволяє закодувати варіант АП як для загальної задачі структурної оптимізації і тим самим позбутися складної структури хромосоми з рис. 5.8.

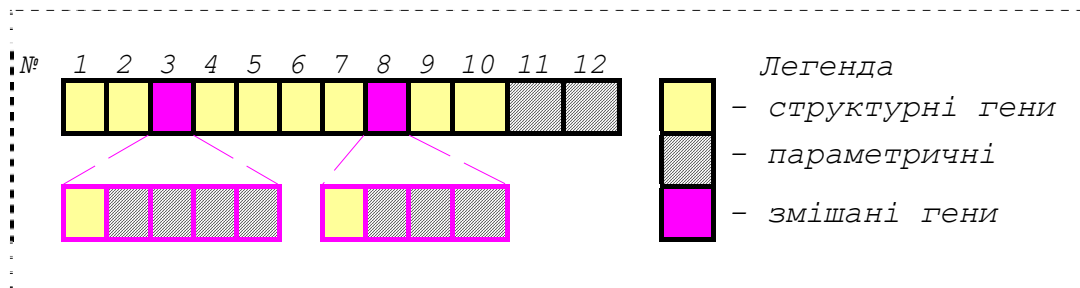


Рис. 5.8. Кодування хромосоми для задачі структурно-параметричної оптимізації АП

### 5.3. Мутація та схрещення хромосом

Мутація – це операція випадкової зміни значень одного чи декількох генів хромосоми. Вона розширює область пошуку, тим самим забезпечує вихід ГА з локальних оптимумів. Схрещення – це операція, в результаті якої отримуються дві хромосоми-нащадки, що спадкують властивості хромосом-батьків. Під час оптимізації вона відповідає етапу експлуатації кращих розв’язків.

При оптимізації АП будемо використовувати звичайну одногенну мутацію [139]. Вона полягає в випадковій зміні значення одного гена обраної хромосоми (рис. 5.9). Хромосома та ген, які потрапляють під мутацію, обирають випадково.

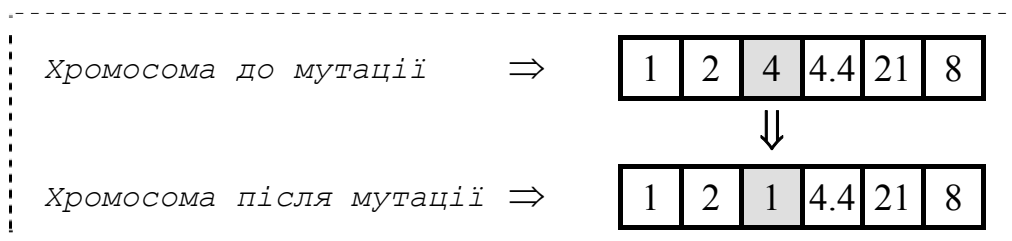


Рис. 5.9. Мутація однорядкових хромосом

Мутацію, що припадає на змішаний ген хромосоми з вкладеною структурою (рис. 5.9), пропонується проводити у такий спосіб:

```

< Згенерувати випадкове число  $\delta \in (0, 1]$  >
If  $\delta > \delta^*$ 
    <Змутувати один параметричний елемент змішаного гена>
Else
    <Згенерувати нові значення всіх елементів змішаного гена >
End.

```

Константа  $\delta^* \in [0, 1]$  задає частоту мутації структурного елемента змішаного гена. Якщо  $\delta^* = 0$ , тоді мутують лише параметричні еле-

менти. Рис. 5.10 ілюструє два варіанти мутації змішаного гена хромосоми з вкладеною структурою.

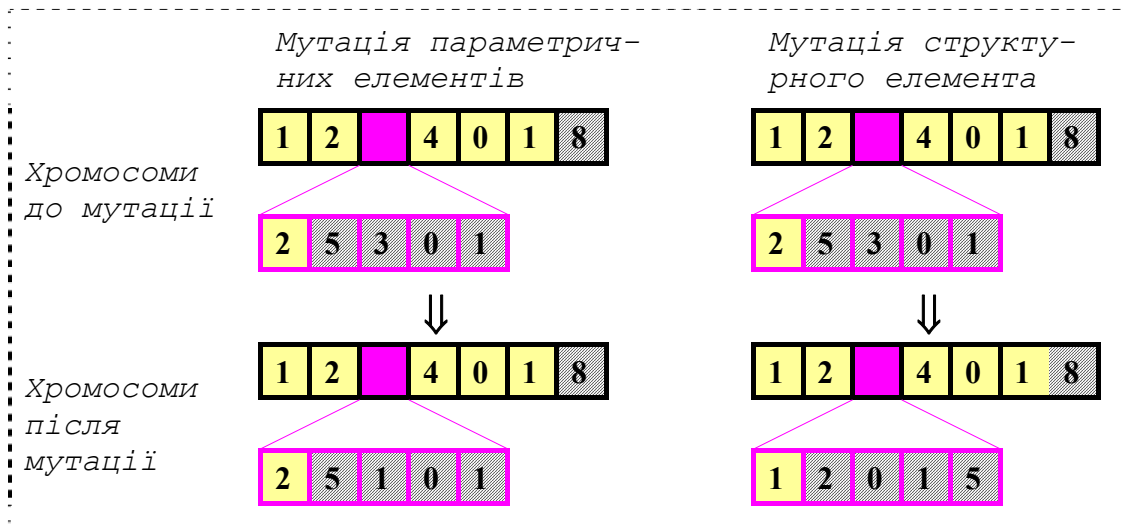


Рис. 5.10. Мутація змішаного гена хромосоми з вкладеною структурою

Схрещення виконуватимемо окремо для структурних, параметричних та змішаних генів. Хромосоми із структурних генів та із параметричних генів з дискретними значеннями пропонується схрещувати за типовою схемою з однією лінією розтину (рис. 5.11). Хромосоми із параметричних генів з неперервними значеннями пропонується схрещувати за типовою схемою на основі лінійної комбінації (рис. 5.12).

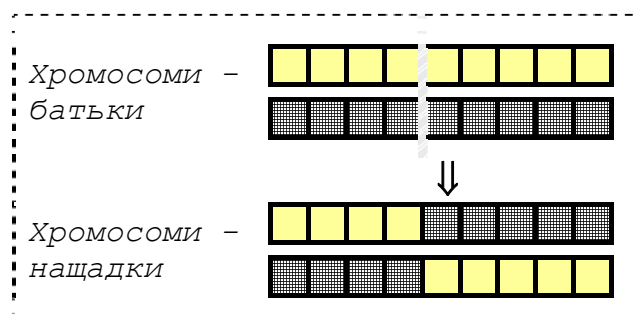


Рис. 5.11. Типове схрещення хромосом з дискретними значеннями генів

Хромосоми з вкладеною структурою пропонується схрещувати в такий спосіб. Якщо структурні елементи змішаного гена батьківських хромосом ідентичні, тоді обидві дочірні хромосоми успадкують їх. При цьому параметричні елементи дочірніх хромосом утворимо за правилами з рис. 5.11 та 5.12. Якщо структурні елементи батьківських хромосом різні, тоді одна дочірня хромосома повністю успадковує

змішаний ген першої батьківської хромосоми, а інша – успадковує змішаний ген другої батьківської хромосоми.

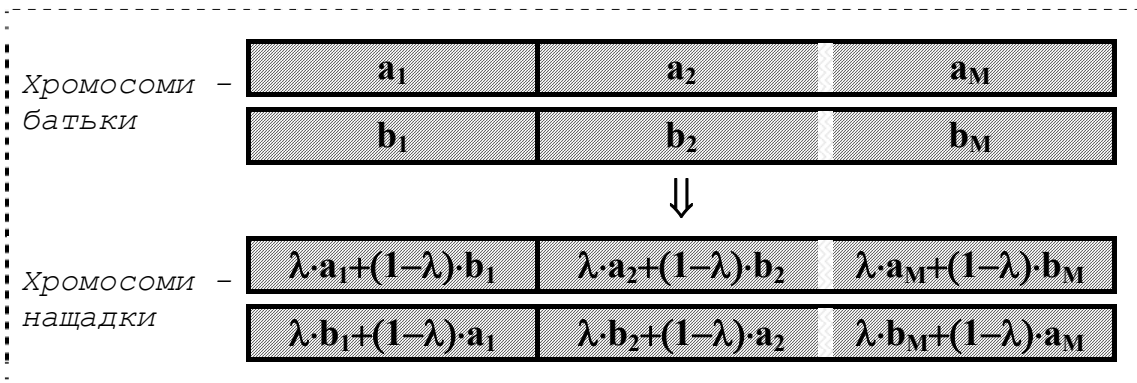


Рис. 5.12. Типове схрещення хромосом з неперервними значеннями генів,  $\lambda \in (0, 1)$

В задачах оптимізації з обмеженням з кількості контрольних процедур мутацію та схрещення виконуватимемо з такою додатковою процедурою: якщо кількість контролів перевищує допустиме значення  $S^*$ , тоді зменшимо алелі одного чи декількох генів. Порядкові номери цих генів визначимо випадково. В задачах загальної структурної оптимізації АП з розширеним кодуванням хромосом (рис. 5.5) порушення генотипу відкоригуємо зміною алелів генів, які не потрапили під мутацію.

#### 5.4. Фітнес-функція хромосом та селекція нової популяції

В задачах максимізації показника надійності фітнес-функції поставимо у відповідність цільову функцію задачі оптимізації. Для задач мінімізації як фітнес-функцію оберемо функцію, що обернена до цільової функції.

З 25087 можливих постановок задач оптимізації надійності АП лише 399 є безумовними. Врахування обмежень задач умовної оптимізації можна здійснити трьома способами [139]: корегуванням, вилученням та штрафуванням.

За *першим способом*, гени хромосоми з недопустимої області корегують, щоб вони задовольняли усі обмеження. Цей спосіб врахування обмежень доцільний для простих правил корегування генів. Наприклад, розглянемо задачу розстановки контрольних точок з обмеженнями на максимальну кількість контролів, вартість та тривалість АП. Для неї правило корегування хромосоми може бути таким: ітераційно випадковим чином обрати ген та занулити його, доки не будуть

виконанні усі обмеження. Корегувати хромосоми слід при їх появі, тобто при ініціалізації початкової популяції, схрещенні та мутації. Якщо в обмеження входять «конфліктні» показники, наприклад, безпомилковість та швидкодія, тоді правила корегування генів стають складними, призводячи до сповільнення оптимізації. При врахуванні всіх обмежень процес повернення хромосоми в допустиму область може наблизитися за складністю до пошуку оптимуму.

За другим способом недопустимі розв'язки вилучають, допускаючи до селекції лише хромосоми, що задовольняють усі обмеження. Цей простий спосіб врахування обмежень можна застосовувати до довільної задачі умовної оптимізації АП. Але вилучення всіх недопустимих розв'язків може призвести до суттєвого уповільнення оптимізації. Наприклад, можлива ситуація, коли всі хромосоми популяції є з області недопустимих розв'язків. Тоді їх усіх треба вилучити і згенерувати нову популяцію, тобто розпочати оптимізацію «з нуля». Крім цього, з теорії математичного програмування відомо, що інколи оптимум знаходять набагато швидше, коли дозволено крокувати по області недопустимих розв'язків. Оптимум часто знаходиться на границі області допустимих розв'язків, тому щоб накрити його «сіткою» з хромосом популяція має містити як допустимі, так і недопустимі розв'язки (рис. 5.13). Щодо оптимізації надійності АП, то вилучення недопустимих хромосом доцільне для задач типу (3.11). Обмеження цих задач визначають зону, поза якою нестача одних показників не компенсується надлишком інших. Іншими словами, «торг» між критеріями поза зоною недоречний, тому доцільно з популяції вилучати недопустимі розв'язки.

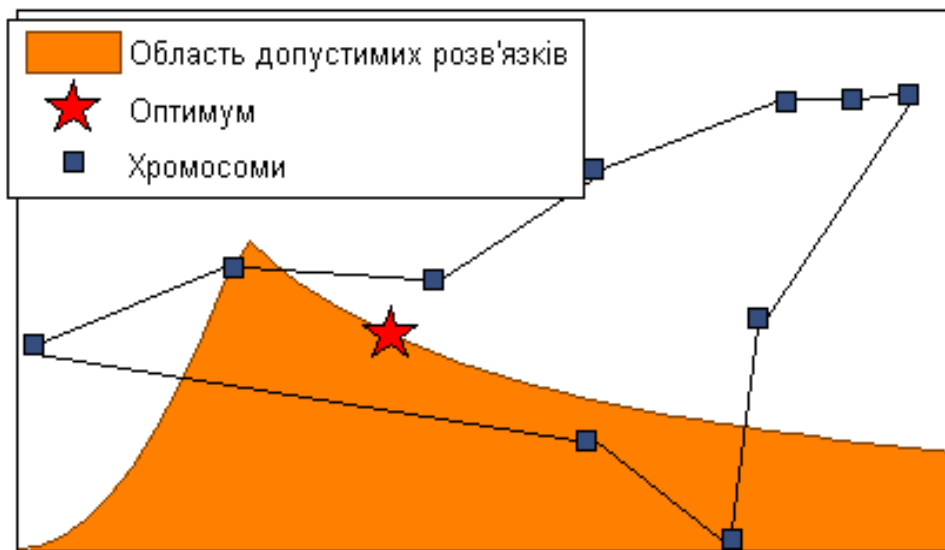


Рис. 5.13. Накриття оптимуму популяцією хромосом



За *третьім способом* при виході хромосоми за допустиму область її штрафують, зменшуючи значення фітнес-функції. Величина штрафу залежить від кількості порушень та відстані від зони допустимих розв'язків. Врахування обмежень за допомогою штрафів доцільне для задач з оптимізацією за одним головним критерієм на кшталт (3.5) – (3.10). Для (3.11) такий підхід недоречний, тому що цільова функція задачі оптимізації являє собою штрафну функцію за порушення мети. Зону, в якій ця штрафна функція діє, визначає система обмежень задачі (3.11).

Будемо використовувати селекцію з елітизмом, коли спочатку в нову популяцію обирається найкраща хромосома з максимальною фітнес-функцією. Якщо фітнес-функція поєднує цільову та штрафну функції, тоді додатково до першою елітної хромосоми в нову популяцію оберемо одну найкращу хромосому з області допустимих розв'язків. Відбір решти хромосом можна проводити за різними схемами, серед яких, на нашу думку, найбільш вдалими є пропорційна, відсічна, турнірна та комбінована селекції.

За *пропорційною селекцією* (селекцією за «колесом рулетки») кожній хромосомі на віртуальному колесі рулетки відповідає свій сектор, площа якого пропорційна значенню її фітнес-функції [139]. Для вибору хромосоми в нову популяцію необхідно кинути кульку на колесо рулетки і визначити на якому секторі вона спиниться. Імітаційно відбір реалізують через розрахунок кумулятивних ймовірностей хромосом та генерацію випадкових чисел з діапазону  $[0, 1]$ . При наближенні стагнації, коли усі хромосоми приблизно однакові, пропорційна селекція майже не відокремлює добрі розв'язки від поганих. Для їх дистанціювання в теорії ГА [139, 144, 156] застосовують нелінійне масштабування цільової функції або ранжування хромосом. Якщо у фітнес-функцію входить штрафна функція, тоді використовують адаптивні штрафи, значення яких залежать не тільки від величини порушення обмежень, але і від якості поточної популяції або кількості виконаних ітерацій ГА.

За *відсічною селекцією* нова популяція формується з  $\tau$ -частки кращих хромосом, які мають однакову ймовірність бути обраними,  $\tau \in (0, 1)$  [144, 156]. Для цього потрібно відсортувати хромосоми за значенням фітнес-функції, що при великих популяціях вимагатиме значного часу. Ще одним недоліком є те, що в обрізаній популяції кращі хромосоми не мають преференцій з виживання.

За *турнірною селекцією* нова популяція формується за результатами турнірів серед невеликого числа випадково обраних хромосом. Переможців турнірів визначають за найбільшим значенням фітнес-функції [144]. За обчислювальною складністю турнірна селекція є

найкращою.

За турнірної селекції можна легко врахувати обмеження задачі оптимізації і без коефіцієнтів штрафної функції за такою *новою* схемою. Переможцем турніру оберемо ту хромосому, що порушує менше обмежень. Якщо кількість порушень обмежень кількома конкуруючими хромосомами однокова, тоді переможець обирається за найкращим значенням цільової функції. Така схема селекції та врахування обмежень задачі оптимізації має добру обчислювальну складність та не вимагає настроювання штрафних коефіцієнтів. Вона краще працюватиме при великому числі обмежень оптимізаційної задачі.

*Комбінована селекція* поєднує відсічну селекцію з іншими. Спочатку залишають  $\tau$ -частку кращих хромосом, а потім застосовують пропорційну або турнірну селекцію.

## **5.5. Гібридна генетична оптимізація надійності алгоритмічних процесів**

В підрозділі пропонуються шляхи поєднання ГА з іншими методами для пришвидшення оптимізації надійності АП. Розглядається гібридизація ГА: 1) з пожадливіми методами дискретної оптимізації для генерування якісної початкової популяції хромосом; 2) з мурашиними алгоритмами оптимізації для покращення поточної популяції за острівною схемою; 3) з методами локального пошуку для покращення окремих хромосом.

### **5.5.1. Генерування якісної початкової популяції за ідеями пожадливих методів дискретної оптимізації**

#### **5.5.1.1. Основна ідея**

Початкову популяція хромосом генерують, як правило, випадковим чином. Для наближення початкової популяції до точки оптимуму нижче пропонується генерувати початкові значення генів не за рівномірним ймовірнісним розподілом, а за таким, що враховує апріорні знання про якість алелів. Кожній алелі поставимо у відповідність деякий показник ефективності АП. Вважатимемо, що чим більше значення показника ефективності алелі, тим більше у неї шансів потрапити в оптимальний розв'язок. Відповідно, при ініціалізації початкової популяції ймовірності вибору алелі пов'яжемо прямою пропорційною залежністю з показниками ефективності відповідних альтернатив. Нижче розглядається як створити ймовірнісний розподіл значень генів для задач структурної оптимізації АП.

### 5.5.1.2. Генерування якісної початкової популяції при виборі способів реалізації операторів та логічних умов

За ідеями пожадливих алгоритмів розв'язання задачі про рюкзак [154] та їх аналогів – градієнтних алгоритмів оптимізації надійності [79, 100], як показник ефективності елемента АП виберемо відношення корисності до витрат. Формули розрахунку ефективності операторів та логічних умов для різних постановок задач зведені в табл. 5.1. В формулах ефективності логічної умови ймовірності помилок типу «хибна тривога» не враховані. Таке спрощення є допустимим при оптимізації АП, в яких такі помилки не завдають великої шкоди та трапляються рідко. Крім того, при формуванні табл. 5.1 вважалось, що при оптимізації багатовимірних АП показник  $p^{0j}(\mathbf{X})$  виступає як критерій оптимізації.

Таблиця 5.1  
Показник ефективності елемента АП при різних постановках задачі оптимізації ( $\eta \in (0, 1)$  – ваговий коефіцієнт)

Критерій та обмеження задачі оптимізації АП	Ефективність оператора $A$	Ефективність логічної умови $\omega$
$p^1(\mathbf{X}), C(\mathbf{X})$	$\gamma = \frac{p_A^1}{c_A}$	$\gamma = \frac{k_\omega^{00}}{c_\omega}$
$I(\mathbf{X}), C(\mathbf{X})$		
$p^1(\mathbf{X}), T(\mathbf{X})$	$\gamma = \frac{p_A^1}{t_A}$	$\gamma = \frac{k_\omega^{00}}{t_\omega}$
$I(\mathbf{X}), T(\mathbf{X})$		
$p^1(\mathbf{X}), C(\mathbf{X}), T(\mathbf{X})$	$\gamma = \frac{p_A^1}{\eta t_A + (1 - \eta)c_A}$	$\gamma = \frac{k_\omega^{00}}{\eta t_\omega + (1 - \eta)c_\omega}$
$I(\mathbf{X}), C(\mathbf{X}), T(\mathbf{X})$		
$p^{0j}(\mathbf{X}), C(\mathbf{X})$	$\gamma = \frac{p_A^{0j}}{c_A}$	$\gamma = \frac{k_\omega^{00j}}{c_\omega}$
$p^{0j}(\mathbf{X}), T(\mathbf{X})$	$\gamma = \frac{p_A^{0j}}{t_A}$	$\gamma = \frac{k_\omega^{00j}}{t_\omega}$
$p^{0j}(\mathbf{X}), C(\mathbf{X}), T(\mathbf{X})$	$\gamma = \frac{p_A^{0j}}{\eta t_A + (1 - \alpha)c_A}$	$\gamma = \frac{k_\omega^{00j}}{\eta t_\omega + (1 - \eta)c_\omega}$

Нехай деякий ген набуває одне з  $N$  альтернативних значень, ефективності яких дорівнюють  $\gamma_1, \gamma_2, \dots, \gamma_N$ . Тоді частоту появи  $i$ -го аль-

тернативного значення цього гена при ініціалізації популяції визначимо таким чином:

$$\beta^i = \frac{\gamma_i}{\sum_{j=1, N} \gamma_j}, \quad i = \overline{1, N}. \quad (5.1)$$

Якщо серед значень гена існує домінуюча альтернатива з великою частотою (5.1), наприклад,  $\beta > 0,95$ , тоді диверсифікація початкової популяція буде поганою. Це порушить важливий принцип метавристичної оптимізації, а саме, дотримання коректного балансу між етапами експлуатації потенційно-найкращої альтернативи та розширенням області пошуку [123, 156]. Для  $\Delta \tau(d_i^j)$  розширення пошукової області встановимо порогове значення ( $\beta^*$ ) частоти появи домінуючої альтернативи. Ця умова аналогічна обмеженню обсягу віртуальних феромонів з максмінного мурашиного алгоритму [175] дискретної оптимізації. Частоти появи решти альтернатив збільшимо за таким алгоритмом:

$$\left. \begin{array}{l} \beta_{\max} = \max_{i=1, N} (\beta^i); \\ \mathbf{If} \quad \beta_{\max} > \beta^* \\ \quad \Delta = (\beta_{\max} - \beta^*) / (N - 1); \\ \quad \beta^i = \beta^i + \Delta, \quad i = \overline{1, N}; \\ \quad k = \arg \max_{i=1, N} (\beta^i); \\ \quad \beta^k = \beta^*; \\ \mathbf{end} \end{array} \right\}, \quad (5.2)$$

де  $\beta^* > 0,5$  – найбільше дозволене значення частоти появи альтернативи в початковій популяції.

### 5.5.1.3. Генерування якісної початкової популяції при оптимізації контролю

Як показник ефективності контролю оберемо його градієнт, тобто відношення приросту безпомилковості АП до витрат через встановлення контролю. Формули градієнта контролю зведені в табл. 5.2. Перші дві формули відповідають градієнту контролю з робіт по оптимізації АП за ймовірнісно-часовими та ймовірнісно-вартісними критеріями [79]. Решта формул узагальнює його на інший перелік показників надійності.

Таблиця 5.2

Гradient  $i$ -го контролю при різних постановках задачі оптимізації АП  
( $\eta \in (0, 1)$  – ваговий коефіцієнт)

Критерій та обмеження задачі оптимізації АП	Gradient $i$ -го контролю
$p^1(\mathbf{X}), C(\mathbf{X})$	$\gamma_i = \frac{\Delta p_i^1(\mathbf{X})}{\Delta C_i(\mathbf{X})} = \frac{p^1(\mathbf{X} _{x_i=1}) - p^1(\mathbf{X} _{x_i=0})}{C(\mathbf{X} _{x_i=1}) - C(\mathbf{X} _{x_i=0})}$
$p^1(\mathbf{X}), T(\mathbf{X})$	$\gamma_i = \frac{\Delta p_i^1(\mathbf{X})}{\Delta T_i(\mathbf{X})} = \frac{p^1(\mathbf{X} _{x_i=1}) - p^1(\mathbf{X} _{x_i=0})}{T(\mathbf{X} _{x_i=1}) - T(\mathbf{X} _{x_i=0})}$
$p^1(\mathbf{X}), C(\mathbf{X}), T(\mathbf{X})$	$\gamma_i = \frac{\Delta p_i^1(\mathbf{X})}{\eta \Delta T_i(\mathbf{X}) + (1 - \eta) \Delta C_i(\mathbf{X})}$
$p^{0j}(\mathbf{X}), C(\mathbf{X})$	$\gamma_{ij} = \frac{1 - \Delta p_i^{0j}(\mathbf{X})}{\Delta C_i(\mathbf{X})} = \frac{1 - (p^{0j}(\mathbf{X} _{x_i=1}) - p^{0j}(\mathbf{X} _{x_i=0}))}{C(\mathbf{X} _{x_i=1}) - C(\mathbf{X} _{x_i=0})}$
$p^{0j}(\mathbf{X}), T(\mathbf{X})$	$\gamma_{ij} = \frac{1 - \Delta p_i^{0j}(\mathbf{X})}{\Delta T_i(\mathbf{X})} = \frac{1 - (p^{0j}(\mathbf{X} _{x_i=1}) - p^{0j}(\mathbf{X} _{x_i=0}))}{T(\mathbf{X} _{x_i=1}) - T(\mathbf{X} _{x_i=0})}$
$p^{0j}(\mathbf{X}), C(\mathbf{X}), T(\mathbf{X})$	$\gamma_{ij} = \frac{1 - \Delta p_i^{0j}(\mathbf{X})}{\eta \Delta T_i(\mathbf{X}) + (1 - \eta) \Delta C_i(\mathbf{X})}$

Примітка: при оптимізації багатовимірних АП  $p^{0j}(\mathbf{X})$  виступає як критерій оптимізації.

## 5.5.2. Покращення популяції хромосом мурашиними алгоритмами за острівною схемою

### 5.5.2.1. Основна ідея

Острівна схема являє собою розв'язання однієї і тієї ж задачі оптимізації одночасно різними методами з обміном найкращими поточними розв'язками через деякий час. Вона часто застосовується для паралельного пошуку екстремуму мережею комп'ютерів. Кожен комп'ютер вирішує задачу оптимізації певним методом і знайдені проміжні розв'язки через локальну мережу передає іншим

комп'ютера. Можлива реалізація острівної схеми і на одному комп'ютері, коли кожному методу оптимізації виділяються деякі обчислювальні ресурси. Застосування острівної схеми дозволяє в простий спосіб поєднати переваги різних методів оптимізації. При цьому методи оптимізації не зазнають змін; потрібно лише реалізувати інтерфейс обміну розв'язками.

Одним із найперспективніших методів дискретної оптимізації є мурашині алгоритми, які ґрунтуються на принципах поведінки колонії біологічних мурах. Мурашині алгоритми забезпечують добрий баланс між швидкістю та точністю оптимізації для різноманітних NP-складних задач [108, 123, 127, 129], в тому числі і для багатовимірної задачі про рюкзак [117, 133, 152, 174]. Між задачами структурної оптимізації надійності АП та задачею про рюкзак встановлено чітку аналогію. Звідси випливає перспективність розробки мурашиного алгоритму структурної оптимізації надійності АП.

Взаємодію між ГА та мурашиним алгоритмом реалізуємо за острівною схемою (рис. 5.14), яку запропоновано для задачі оптимальної маршрутизації вантажівок [162]. Перспективність такої схеми для різноманітних задач дискретної оптимізації підтверджено подальшими дослідженнями інших науковців в роботах [116, 159, 182].

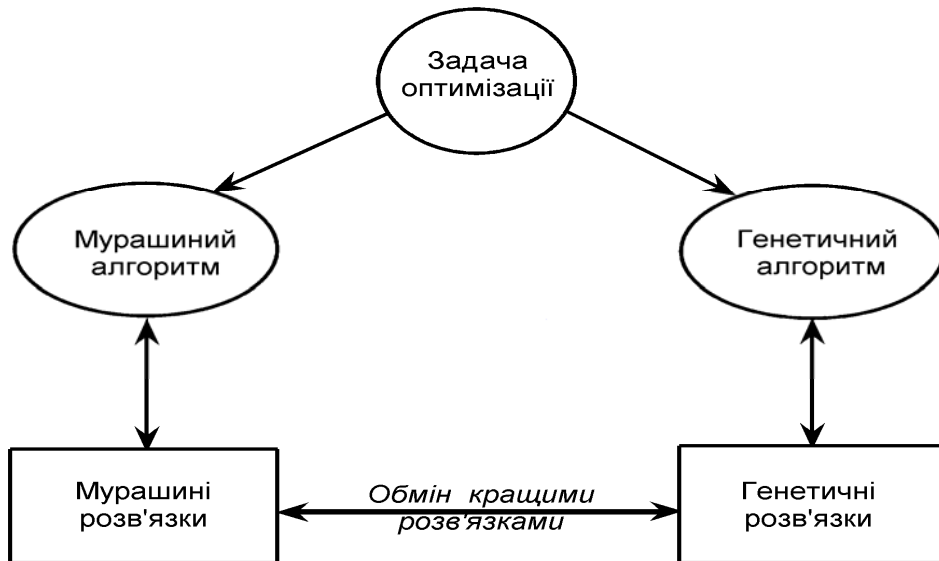


Рис. 5.14. Генетико-мурашина оптимізація за острівною схемою

У випадку однокомп'ютерної реалізації оптимізації генетичний та мурашиний алгоритми працюють послідовно протягом декількох ітерацій. Якщо ГА покращив розв'язок, тоді він передається мурашиному алгоритму для оновлення феромонних слідів. Далі оптимізація йде за мурашиним алгоритмом і найкращий розв'язок передається до ГА, де він заміщує найгіршу хромосому. Після цього знову оптимізуємо за ГА і т.д. Кількість ітерацій ГА та мурашиного алгоритму мож-

на поставити в залежність від якості оптимізації. Таку адаптацію пропонується проводити за принципом «дворукого бандита» [68, 145]. Ідея адаптації полягає в збільшенні кількості ітерацій для кращого алгоритму оптимізації, та відповідно зменшення тривалості роботи гіршого алгоритму. Переможця в змаганні алгоритмів визначатимемо за збільшенням фітнес-функції на одну ітерацію алгоритму. Корегувати кількість ітерацій можна як за результатами кожного включення, так і за сумарною ефективністю за декілька включень.

Таким чином, в схемі на рис. 5.14 відомими є ГА та інтерфейс взаємодії алгоритмів. Залишилося розробити мурашиний алгоритм оптимізації надійності АП. Мурашині алгоритми придатні для розв'язання задач дискретної оптимізації. Серед задач оптимізації АП дискретними є задачі структурної оптимізації надійності АП, які мають порядкові номерами 1–5 за класифікацією з табл. 3.1. Щоб довільну задачу дискретної оптимізації вирішити мурашиними алгоритмами необхідно звести її до пошуку шляху на деякому графі, обрати способи ініціалізації та оновлення феромону, та визначити евристичні правила вибору маршруту. Нижче пропонуються реалізації основних складових мурашиної оптимізації надійності АП.

### 5.5.2.2. Графове представлення задач структурної оптимізації алгоритмічних процесів

Позначимо через  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  – вектор керованих змінних задачі структурної оптимізації надійності АП. Кожна керована змінна  $x_i$  може набувати значення з дискретної множини  $\mathbf{D}_i = \{d_{i1}, d_{i2}, \dots, d_{iu_i}\}$ , елементи якої відповідають реалізаціям оператора чи логічної умови АП, тобто  $x_i \in \mathbf{D}_i, i = \overline{1, n}$ . Кожній реалізації поставимо у відповідність одну вершину графа. Такий підхід відповідає ідея мурашиного розв'язання задачі про рюкзак [117, 133, 152, 174], коли кожному предмету відповідає одна вершина графа. Для задач структурної оптимізації АП аналогами предметів є варіанти реалізацій операторів та логічних умов.

При мурашиному розв'язанні задачі про рюкзак можливі 3 варіанти розміщення віртуальних феромонів на графі:

- 1) феромони залишаються на вершинах графа [152];
- 2) феромони залишаються на дугах, що з'єднують вершини з множини  $\mathbf{D}_i$  з вершинами з множини  $\mathbf{D}_{i+1}, i = \overline{1, n-1}$  [133];
- 3) феромони залишаються на дугах, що з'єднують вершини з множини  $\mathbf{D}_i$  з вершинами з множини  $\mathbf{D}_j, i, j = \overline{1, n}$  [174].

Обчислювальні експерименти, які проведені в [117] свідчать, що

найкращі розв’язки знаходять за третім варіантом розміщення феромонів. Щодо динаміки оптимізації, то найкращим є перший варіант, який поступається за цим показником лише при наближенні до оптимуму. За острівної схеми ми найбільш зацікавлені в добрій динаміці оптимізації, тобто в швидкому генеруванні розв’язків, які близькі до оптимального. Перетворити субоптимальні розв’язки, які знайдені одним методом, на оптимальні зможе другий метод. Крім того, на заключному етапі оптимізації для швидкого досягнення екстремуму можна застосувати методи локального пошуку. Враховуючи і те, що програмна реалізація мурашиної оптимізації за першим варіантом найпростіша, віртуальні феромони розміщуватимемо на вершинах графа. Включимо до графа і одну фіктивну вершину, з якою пов’яжемо всі інші. Задача вибору варіанта АП зведемо до пошуку дерева, яке включає фіктивну вершину та по одній вершини з кожної множини  $D_i$ ,  $i = 1, n$  (рис. 5.15). Найкращим розв’язком буде дерево графа, яке задовольняє постановку задачі оптимізації з підрозділу 3.2.

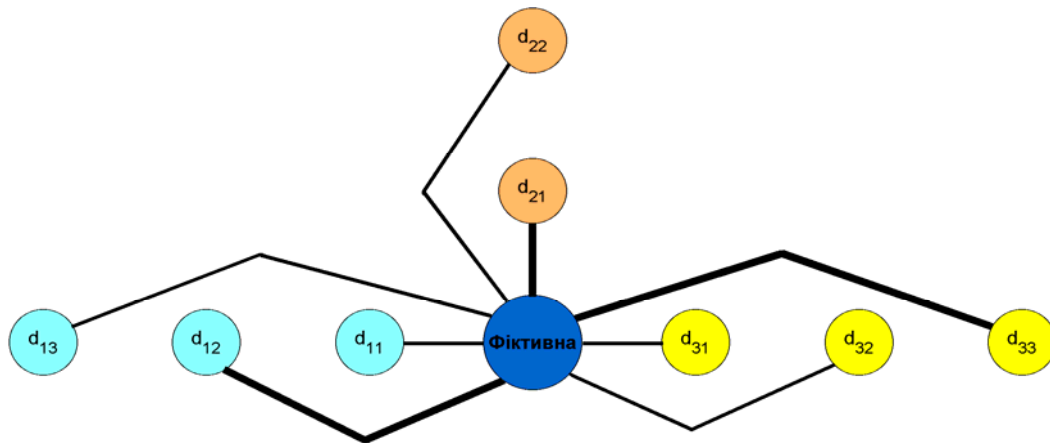


Рис. 5.15. Графове представлення задачі структурної оптимізації АП для трьох керованих змінних (варіант розв’язку виділено жирними лініями)

### 5.5.2.3. Реалізація основних елементів мурашиного алгоритму

Для багатьох тестових задач дискретної оптимізації найкращі розв’язки знайдені максимінним мурашиним алгоритмом [175]. Тому на його базі створимо мурашиний алгоритм оптимізації надійності АП.

Відповідно до [175] встановимо мінімально та максимально можливі обсяги феромонів на кожній вершині графа:

$$\tau(d_{ij}) \in [\tau_{\min}, \tau_{\max}], \quad i = \overline{1, n}, \quad j = \overline{1, u_i}. \quad (5.3)$$



На початку алгоритму на кожен вузол відкладемо  $\tau_{\max}$  кількість феромонів. Кількість феромонів, які відкладаємо на вузли графа за маршрутом  $\mathbf{X}$  визначимо таким чином:

$$\Delta\tau(d_{ij}, \mathbf{X}) = \begin{cases} Q/F(\mathbf{X}), & \text{якщо } d_{ij} \in \mathbf{X}, \\ 0, & \text{інакше} \end{cases}, \quad i = \overline{1, n}, \quad j = \overline{1, u_i}, \quad (5.4)$$

де  $Q$  – константа;

$F(\mathbf{X})$  – значення фітнес-функції з генетичної оптимізації надійності АП для розв'язку  $\mathbf{X}$ .

Після кожної ітерації феромони відкладатимемо тільки на вузлах графа за найкращим поточним маршрутом ( $\mathbf{X}^+$ ). Якщо він гірше найкращого маршруту ( $\mathbf{X}^{best}$ ), знайденого від початку роботи алгоритму, тоді феромони відкладемо і на вузлах маршруту  $\mathbf{X}^{best}$ . Математично запишемо це таким чином:

$$\Delta\tau(d_{ij}) = \begin{cases} \Delta\tau(d_{ij}, \mathbf{X}^+) & \text{якщо } F(\mathbf{X}^+) > F(\mathbf{X}^{best}), \\ \Delta\tau(d_{ij}, \mathbf{X}^+) + \Delta\tau(d_{ij}, \mathbf{X}^{best}) & \text{інакше} \end{cases}, \quad (5.5)$$

де  $\Delta\tau(d_{ij})$  – обсяг феромонів, які відкладаємо на вузол  $d_{ij}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, u_i}$ .

Випаровування феромонів реалізуємо за типовою схемою

$$\tau(d_{ij})^{<t+1>} = (1-p) \cdot \tau(d_{ij})^{<t>}, \quad i = \overline{1, n}, \quad j = \overline{1, u_i}, \quad (5.6)$$

де  $p \in (0, 1)$  – коефіцієнт випаровування феромонів;

$t$  – порядковий номер поточної ітерації мурашиного алгоритму.

З (5.3)–(5.6) отримуємо таке правило оновлення феромонів:

$$\tau(d_{ij})^{<t+1>} = \max \left\{ \tau_{\min}, \min \left( \tau_{\max}, (1-p) \cdot \tau(d_{ij})^{<t>} + \Delta\tau(d_{ij}) \right) \right\}, \quad (5.7)$$

$$i = \overline{1, n}, \quad j = \overline{1, u_i}.$$

Ймовірність включення вузла графа  $d_{ij}$  в маршрут мурахи визначимо за таким випадково-пропорційним правилом:

$$prob(d_{ij}) = \frac{\tau(d_{ij})^\alpha \cdot \gamma(d_{ij})^\beta}{\sum_{s=1, u_i} \tau(d_{is})^\alpha \cdot \gamma(d_{is})^\beta}, \quad i = \overline{1, n}, \quad j = \overline{1, u_i}. \quad (5.8)$$

де  $\gamma(d_{ij})$  – ефективність альтернативи  $d_{i,j}$ , яка розрахована за формулами пункту 5.5.1;

$\alpha > 0$  та  $\beta > 0$  – коефіцієнти, які визначають важливості досвіду популяції (слідів феромону) та пожадливого правила (ефективності альтернатив) при виборі продовження маршруту.

Якщо  $\alpha \rightarrow 0$ , тоді найвірогіднішим буде вибір альтернатив з найбільшою ефективністю. У класичній теорії оптимізації це відповідає, так званому, пожадливому алгоритму. Якщо  $\beta \rightarrow 0$ , тоді працює лише феромонне підсилення, що приводить до швидкого завершення роботи алгоритму через збігання маршрутів усіх мурах до одного субоптимального розв'язку.

Зауважимо, що (5.8) визначає лише ймовірності вибору альтернатив. Сам вибір здійснюється за принципом колеса рулетки, на якій кожній альтернативі відведено сектор з площею пропорційною до ймовірності (5.8).

#### 5.5.2.4. Мурашиний алгоритм оптимізації надійності

Мурашиний алгоритм оптимізації структурної надійності АП пропонується нижче.

```

<Ініціалізація параметрів алгоритму  $\alpha, \beta, Q, \tau_{min}, \tau_{max}$ >
For  $i=1:n$ 
  For  $j=1:u_i$ 
    <Розрахувати  $\gamma(d_{ij})$ >
    < $\tau(d_{ij})=\tau_{max}$ >
  End
End
<Згенерувати допустимий розв'язок  $M^{best}$ >
<Обчислити значення фітнес-функції  $F(M^{best})$ >
%Основний цикл:
For  $t=1:t_{max}$  |  $t_{max}$  - кількість ітерацій алгоритму
  For  $k=1: number\_ants$  | для кожної мурахи
    <Побудувати маршрут  $M^k(t)$  за правилом (5.8)>
    <Обчислити значення фітнес-функції  $F(M^k(t))$ > End
  <Обрати краций поточний розв'язок  $M^+$ >
  <Зафіксувати значення фітнес-функції  $F(M^+)$  розв'язку  $M^+$ >
  <Оновити феромонні сліди за правилом (5.7)>
  If  $F(M^+) > F(M^{best})$ 
    < $M^{best}=M^+$ >  $ma$  < $F(M^{best})=F(M^+)$ >
  EndIf
End
<Вивести найкращий розв'язок  $M^{best}$ >

```

Принципова відмінність запропонованого алгоритму від мурашиних алгоритмів вирішення задачі про рюкзак полягає в різному врахуванні обмежень. В задачі про рюкзак при додаванні будь-якого предмета можна перевірити виконання обмежень задачі оптимізації. Причому, така перевірка є нескладною в обчислювальному плані. Тому в задачі про рюкзак ймовірнісний розподіл на кшталт (3.22) створюється лише для допустимих альтернатив. В загальних задачах структурної оптимізації АП обмеження є нелінійними, тому в більшості випадків неможливо перевірити задоволення обмежень за фрагментами розв'язку. Тому перевірка обмежень здійснюється не для фрагмента, а для цілого розв'язку. Якщо протягом однієї ітерації мурашиного алгоритму не знайдено ні одного допустимого розв'язку, тоді феромони додаються лише на вершини маршруту  $M^{best}$ .

### **5.5.3. Покращення хромосоми методами локального пошуку**

#### **5.5.3.1. Основна ідея**

Результати обчислювальних експериментів з генетичної оптимізації [92, 139, 144, 156] свідчать, що застосування методів локального пошуку суттєво пришвидшує знаходження екстремуму. Особливо відчутним таке пришвидшення стає коли хромосоми знаходяться в околі екстремуму. Тому підключати процедури локального пошуку треба частіше на заключних ітераціях оптимізації. На початку генетичної оптимізації локальний пошук не доцільний.

Нижче розглядається як застосувати методи локального пошуку для скорочення часу генетичної оптимізації надійності АП. Методи локального пошуку розроблені для дискретних або неперервних задач оптимізації. В загальному випадку аналізована задача оптимізації є непервно-дискретною. Тому для застосування неперервних методів локального пошуку слід зафіксувати дискретні алелі хромосом. Відповідно, при застосуванні дискретних методів локального пошуку зафіксуємо неперервні алелі хромосом. Крім того, на відміну від ГА методи локального пошуку зазвичай проблемно-орієнтовані. Вони розроблені для вирішення специфічних задач, наприклад, розстановки контрольних точок, і не придатні для загальних задач структурної оптимізації АП. Тому додатково зафіксуємо ті алелі хромосом, які не можна покращити відповідним методом локального пошуку.

#### **5.5.3.2. Застосування градієнтних методів оптимізації контролю**

Для задач розстановки контрольних точок та вибору кратностей контролів розроблено градієнтні алгоритми оптимізації. Алгоритми з

[75, 76, 79] вирішують пряму та обернену задачу оптимізації з одним обмеженням, що відповідає постановкам (3.5), (3.6) та (3.9). Алгоритми з розділу 4 крім цього враховують і обмеження з ймовірностей появи помилок кожного типу, що відповідає постановкам (3.7), (3.8) та (3.10).

При застосуванні означених алгоритмів як методів локального пошуку слід зафіксувати всі алелі хромосом, окрім тих, що відповідають за кратності контролю. В самі алгоритми оптимізації внесемо такі зміни:

- початковим розв'язком вважатимемо хромосому з ГА, а не відсутність контролю;
- якщо початковий розв'язок порушує ресурсне обмеження, тоді замість відкату на 1 крок зменшуватимемо кількість контрольних точок поки не потрапимо в область допустимих розв'язків.

#### **5.5.3.3. Застосування мурашиних алгоритмів дискретної оптимізації**

Запропонований в пункті 5.5.2.4 мурашиний алгоритм придатний для вирішення як загальної, так і окремих задач структурної оптимізації надійності АП. Тому при застосуванні мурашиного алгоритму як методів локального пошуку зафіксуємо лише значення параметричних генів хромосоми. Передану в мурашиний алгоритм хромосому вважатимемо за перший допустимий розв'язок  $M^{best}$ . Через це вимагатимемо, щоб в мурашиний алгоритм для покращення передавалися лише хромосоми з допустимої області.

#### **5.5.3.4. Застосування класичних методів неперервної оптимізації**

Якщо зафіксувати значення всіх структурних генів та дискретні значення параметричних генів хромосоми, тоді задача оптимізації надійності АП зведеться до задачі математичного програмування. Для її вирішення можна застосовувати добре розроблені методи неперервної оптимізації [69] та відповідні програмні пакети, наприклад, [158].

#### **5.5.3.5. Спільне застосування різних методів локального пошуку**

Для покращення хромосом можна одночасно застосовувати декілька методів локального пошуку за такою схемою:

- 1) фіксуємо значення структурних генів та покращуємо значення параметричних генів методом неперервної оптимізації;

- 2) фіксуємо знайдені значення параметричних генів та покращуємо значення структурних генів мурашиними або пожадли-вими алгоритмами;
- 3) повторюємо попередні кроки поки покращується розв'язок або до вичерпання ліміту часу.

Запропонована схема є аналогом методу покоординатного спуску Гауса–Зейделя при пошуку екстремуму функції багатьох змінних. При цьому кожній координаті відповідає сукупність параметричних або структурних генів.

## **5.6. Експериментальні дослідження з генетичної оптимізації багатовимірних алгоритмічних процесів**

Наводяться результати комп'ютерних експериментів з тестування запропонованих ГА оптимізації надійності АП.

### **5.6.1 Завдання та методика комп'ютерних експериментів**

З 25087 можливих задач оптимізації надійності, експериментальні дослідження проведемо з оптимізації контролю багатовимірних АП. Вибір задач цього типу обумовлено двома причинами. По-перше, оптимізація розташування контролю є однією із найбільш уживаних задач в практиці надійного проектування АП. По-друге, алгоритмічна складність цих задач є достатньою, щоб при тестуванні виявити особливості їх оптимізації. Для цих задач розроблена відповідна бібліотека тестових задач.

Тестувати ГА будемо з метою встановлення якості знайденого розв'язку та витрат часу на оптимізацію. Дослідження проведемо експериментальним шляхом багатократного прогону алгоритмів оптимізації з фіксацією результатів після кожного запуску. Експерименти проведемо для ГА з такими параметрами: розмір популяції – 25 хромосом; кількість схрещень за 1 епоху – 30; кількість мутацій за 1 епоху – 30.

Дослідження точності алгоритмів проведемо за значенням цільової функції для тих розв'язків, що задовольняють усі обмеження задачі оптимізації. Для порівняння як альтернативний метод оптимізації оберемо градієнтний алгоритм оптимізації АП з робіт [79, 86, 88]. При дослідженнях швидкодії алгоритмів засічемо час знаходження оптимуму при кожному прогоні та усереднимо ці значення. При дослідженні впливу схеми селекції на час оптимізації для кожної тестової задачі сформуємо 10 різних початкових популяцій хромосом. Потім для кожної початкової популяції запустимо по 10 раз ГА з кожним типом селекції та усереднимо час оптимізації. Дослідження проведемо

для турнірної, відсічної та пропорційної селекції з елітизмом.

На час оптимізації впливає вибір початкової популяції. Нижче запропонована нова процедура генерування початкової популяції, тому порівнюємо її з відомою – випадковою ініціалізацією. Порівнювати будемо за частотою вгадувань оптимальних значень генів. Також на прикладі однієї з тестових задач дослідимо як впливає якість початкової популяції на швидкість оптимізації.

Обчислювальні експерименти проведемо в середовищі MATLAB 6.5 в операційній системі Windows XP на комп'ютері з оперативною пам'яттю в 256 MB та процесором Intel Celeron 900 МГц.

### 5.6.2 Генетичний розподіл контрольних точок при обмеженнях на безпомилковість

Для оптимізації розподілу контрольних точок будемо використовувати ГА оптимізації з підрозділу 5.1. Операції кодування, схрещування та мутації описано в підрозділах 5.2 та 5.3. Для оцінювання якості розв'язків пропонується така фітнес-функція [86, 110]

$$F(\mathbf{X}) = \begin{cases} \frac{1}{C(\mathbf{X})}, & \text{якщо } \mathbf{X} \text{ – недопустимий розв'язок} \\ \frac{1 - D(\mathbf{X})}{C(\mathbf{X})}, & \text{інакше} \end{cases}, \quad (5.9)$$

де  $D(\mathbf{X}) \in [0, 1]$  – штрафна функція, яка враховує порушення хромосоמוю  $\mathbf{X}$  обмежень задач оптимізації (3.9) та (3.10).

Пропонується використовувати таку штрафну функцію [86, 110]

$$D(\mathbf{X}) = \frac{1}{m+1} \cdot \left( \max \left( 0, \frac{P^* - p^1(\mathbf{X})}{P^*} \right) + \sum_{j=1}^m \left( \frac{\Delta_j(\mathbf{X})}{\Delta_j^{\max}} \right)^\alpha \right), \quad (5.10)$$

де  $\alpha > 0$  – коефіцієнт, що визначає вагомість штрафів за порушення обмежень за ймовірностями помилок  $(q_1, q_2, \dots, q_m)$ ;

$$\Delta_j(\mathbf{X}) = \max \left( 0, p^{0j}(\mathbf{X}) - q_j \right) \text{ – величина порушення хромосо-$$

мою  $\mathbf{X}$   $j$ -го обмеження,  $j = \overline{1, m}$ ;

$$\Delta_j^{\max} = \max_{k=1, \text{pop\_size}} (\varepsilon, \Delta_j(\mathbf{X})) \text{ – якість поточної популяції, яка ви-}$$

значається максимальними порушеннями обмеження по  $q_j, j = \overline{1, m}$ .

Штрафи (5.10) за одні й ті ж порушення обмежень при різних популяціях хромосом неоднакові. Ці штрафи адаптуються під якість популяції таким чином, щоб при селекції дистанціювати погані розв'язки від добрих.

ГА оптимізації розподілу контрольних точок протестований на задачах А\_4\_1 та А\_4\_2. Результати оптимізації наведені в табл. 5.3.

Таблиця 5.3

Результати генетичного РКТ для тестових задач А\_4\_1 та А\_4\_2

задача		$P^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$C$
$n=20$	А_4_1	0,9636	0,0063	0,0086	0,0088	0,0127	206,98
	А_4_2	0,9568	0,0079	0,0094	0,0106	0,0153	156,35
$n=40$	А_4_1	0,9457	0,0095	0,0130	0,0136	0,0182	457,72
	А_4_2	0,9271	0,0129	0,0161	0,0182	0,0257	324,99
$n=60$	А_4_1	0,9308	0,0133	0,0166	0,0186	0,0206	542,61
	А_4_2	0,9014	0,0177	0,0210	0,0256	0,0343	484,31
$n=80$	А_4_1	0,8685	0,0236	0,0305	0,0339	0,0435	718,87
	А_4_2	0,8727	0,0245	0,0303	0,0344	0,0381	704,52
$n=100$	А_4_1	0,8543	0,0257	0,0328	0,0380	0,0492	989,83
	А_4_2	0,8456	0,0293	0,0363	0,0412	0,0475	864,94
$n=120$	А_4_1	0,8381	0,0289	0,0371	0,0422	0,0536	1114,0
	А_4_2	0,8107	0,0356	0,0444	0,0503	0,0590	965,44

Комп'ютерні експерименти свідчать, що тривалість оптимізації за генетичним алгоритмом значно більша ніж при використанні градієнтних алгоритмів з підрозділу 4.4.3 [81]. Для прискорення роботи генетичної оптимізації нижче пропонується [109, 110]: 1) процедура ініціалізації якісної початкової популяції; 2) процедура швидкого обчислення показників надійності АП; 3) ефективна схема селекції.

*Процедура ініціалізації популяції* генерує хромосоми в області допустимих розв'язків або в її околі. В цій процедурі значення гена  $x_i$  пропонується визначати таким чином [109, 110]:

$$x_i = \begin{cases} \frac{f(\gamma_i)}{\xi^{g(\gamma_i)}} & \text{якщо } \xi^{g(\gamma_i)} \geq 0.5 \\ \frac{f(\gamma_i)}{\xi^{g(\gamma_i)}} & \text{якщо } \xi^{g(\gamma_i)} < 0.5 \end{cases}, \quad i = \overline{1, n}, \quad (5.11)$$

де  $\xi$  – випадкове число з діапазону  $[0, 1]$ ;

$\gamma_i$  – градієнт  $i$ -го контролю  $\omega_i$ ;

$$f(\gamma_i) = \frac{\min_{j=1,n}(\gamma_j)}{\gamma_i} \text{ – показник ефективності встановлення гена } x_i \text{ в } 1,$$

$$f(\gamma_i) \in [0, 1];$$

$$g(\gamma_i) = \frac{\gamma_i}{\max_{j=1,n}(\gamma_j)} \text{ – показник ефективності встановлення гена } x_i \text{ в } 0,$$

$$g(\gamma_i) \in [0, 1]$$

Покажемо, що для ефективних контролів формула (5.11) встановлює алелі генів в одиницю, а для неефективних в нулі.

Для ідеального контролю, який не погіршує ймовірнісні характеристики АП,  $\gamma_i \rightarrow \infty$ . Це означає, що  $\lim_{\gamma_i \rightarrow \infty} f(\gamma_i) = 0$  та

$$\lim_{\gamma_i \rightarrow \infty} g(\gamma_i) = \infty. \quad \text{Таким чином} \quad \lim_{\gamma_i \rightarrow \infty} \frac{f(\gamma_i)}{g(\gamma_i)} = \frac{0}{\infty} = 0. \quad \text{Тому}$$

$$\lim_{\gamma_i \rightarrow \infty} \xi^{g(\gamma_i)} = \xi^0 = 1. \text{ Для абсолютно неефективного контролю } \gamma_i \rightarrow 0.$$

Це означає, що  $\lim_{\gamma_i \rightarrow 0} f(\gamma_i) \rightarrow \infty$  та  $\lim_{\gamma_i \rightarrow 0} g(\gamma_i) \rightarrow 0$ . Таким чином

$$\lim_{\gamma_i \rightarrow 0} \frac{f(\gamma_i)}{g(\gamma_i)} = \frac{\infty}{0} = \infty. \text{ Тому } \lim_{\substack{\gamma_i \rightarrow 0, \\ 0 \leq \xi \leq 1}} \xi^{g(\gamma_i)} = \lim_{\substack{\gamma_i \rightarrow 0, \\ 0 \leq \xi \leq 1}} \xi^\infty = 0.$$

На основі формули (5.11) пропонується така процедура ініціалізації популяції:

*⟨Ініціалізувати розмір популяції pop\_size⟩;*

**For**  $i=1 \dots n$  | для кожного гена

*⟨Обчислити за формулою (4.19) градієнт контролю  $\gamma_i(x_i)$ ⟩;*

*⟨Обчислити показники  $f(\gamma_i)$ ,  $g(\gamma_i)$ ⟩;*

**For**  $j=1 \dots pop\_size$  | для кожної хромосоми

*⟨Згенерувати випадкове число  $\xi$  з діапазону  $[0, 1]$ ⟩;*

*⟨Ініціалізувати за формулою (5.11) значення  $x_i$  в хромосомі  $\mathbf{X}^j$ ⟩;*

**End;**

**End.**

Порівняємо ефективність запропонованої процедури ініціалізації початкової популяції з випадковою ініціалізацією, яка



використовується в ГА. Ефективність будемо оцінювати частотою вгадування оптимальних значень генів за такою формулою

$$w = \frac{1}{pop\_size \cdot n} \cdot \sum_{j=1}^{pop\_size} \sum_{i=1}^n \Delta(x_i^{opt}, x_i^j), \quad (5.12)$$

$$\text{де } \Delta(x_i^{opt}, x_i^j) = \begin{cases} 1, & x_i^{opt} = x_i^j \\ 0, & x_i^{opt} \neq x_i^j \end{cases};$$

$x_i^{opt}$  – алель  $i$ -го гену оптимальної хромосоми  $\mathbf{X}^{opt}$ ;

$x_i^j$  – алель  $i$ -го гену  $j$ -ої згенерованої хромосоми  $\mathbf{X}^j$ ,  
 $j = 1, pop\_size$ .

Показник  $w$  характеризує середню частоту збігу алелів генів хромосом початкової популяції з алелями генів оптимальної хромосоми. Великі значення частоти свідчать, що ініціалізовані хромосоми наближаються до оптимальних.

В табл. 5.4 порівнюється частота вгадувань оптимальних генів при випадковій та запропонованій процедурах ініціалізації для тестових задач А\_4\_1 та А\_4\_2.

Таблиця 5.4

Середня частота вгадувань  $w$   
(для 50 експериментів, коли  $pop\_size=50$ )

Задача		$n=20$	$n=40$	$n=60$	$n=80$	$n=100$	$n=120$	Середня
А_4_1	випадкова ініціалізація	0,49	0,47	0,49	0,50	0,50	0,51	0,49
	запропонована ініціалізація	0,65	0,63	0,62	0,71	0,65	0,67	0,65
А_4_2	випадкова ініціалізація	0,48	0,49	0,50	0,50	0,49	0,49	0,49
	запропонована ініціалізація	0,72	0,69	0,68	0,67	0,64	0,65	0,68

На рис. 5.16 порівнюються розподілення часу генетичної оптимізації при запропонованій та випадковій процедурах ініціалізацій популяції для задачі А\_4\_2 ( $n=20$ ). Рис. 5.16 свідчить, що запропонована процедура ініціалізації популяції прискорює генетичну оптимізацію більше ніж в 2 рази.

Процедура швидкого обчислення показників надійності АП. На рис. 5.17а зображено результат профілювання генетичного розподілу контрольних точок для тестової задачі А\_4\_2  $n=20$ . З цього рисунку

видно, що найбільше часу витрачається на обчислення показників надійності АП за формулами (4.2)–(4.5). Щоб прискорити обчислення, спочатку розрахуємо значення  $P_{A_i}(x_i)$  та  $c_{A_i}(x_i)$  для всіх можливих значень  $x_i$ ,  $i = \overline{1, n}$  та збережемо їх в оперативній пам'яті. Тоді, під час оптимізації розрахунок показників надійності варіантів АП здійснюється за формулами (4.4), (4.5) для відповідних наперед розрахованих значень  $P_{A_i}(x_i)$  та  $c_{A_i}(x_i)$ .

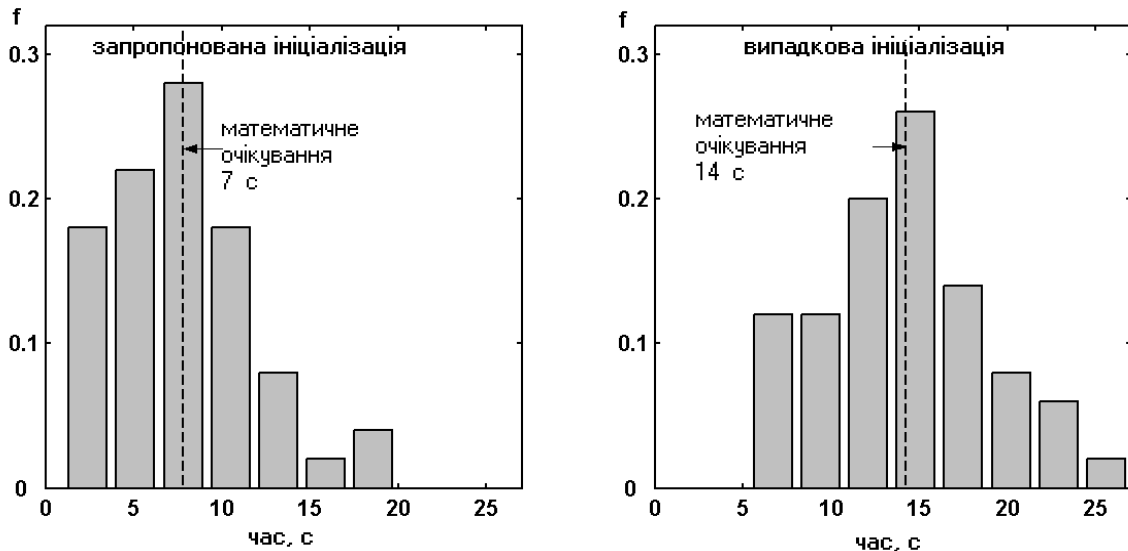


Рис. 5.16. Порівняння запропонованої і типової процедур ініціалізації популяції для задачі А\_4\_2 при  $n=20$  (статистика 50 обчислювальних експериментів)

На рис. 5.17б зображено результати профілювання генетичного розподілу контрольних точок з швидким обчисленням показників надійності АП. З цього рисунку видно, що запропонований спосіб обчислення показників надійності дозволяє значно скоротити час оптимізації.

*Вибір ефективної схеми селекції.* На рис. 5.18–5.20 порівнюються середні динаміки генетичної оптимізації розподілу контрольних точок з різними стратегіями селекцій. Всі схеми селекцій були протестовані на 5 різних початкових популяціях. Для кожної початкової популяції ГА оптимізації запускався 25 разів. На рис. 5.18 зображено динаміку оптимізації при різних порогах  $\tau$  відсічної селекції. Параметр  $\tau$  показує процент хромосом, яких вилучають з популяції до застосування колеса рулетки. Добрі результати досягаються при  $\tau \approx 50\%$ . На рис. 5.19 зображено динаміку оптимізації задач при різній кількості  $t$  учасників змагань турнірної селекції. Добрі результати досягаються при  $t \approx 6$ .

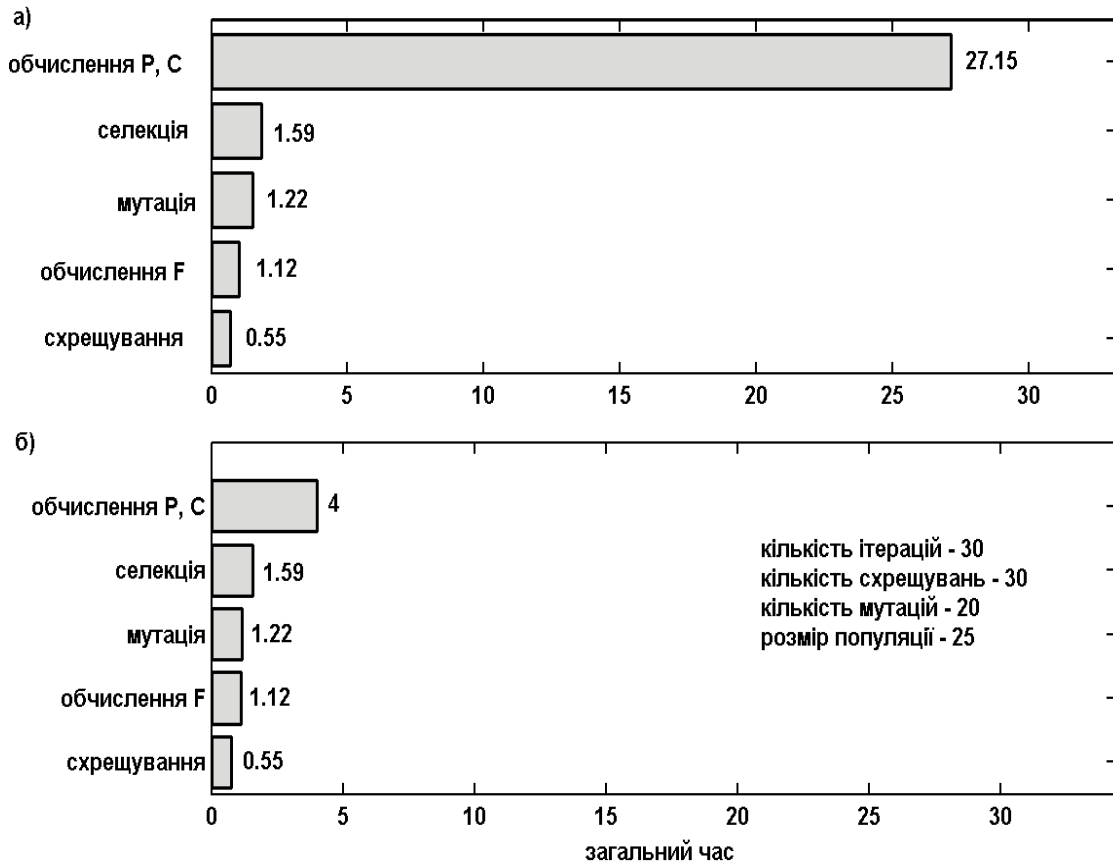


Рис. 5.17. Результати профілювання генетичної РКТ з обчисленням показників надійності АП:  
а) звичайним способом; б) швидким способом

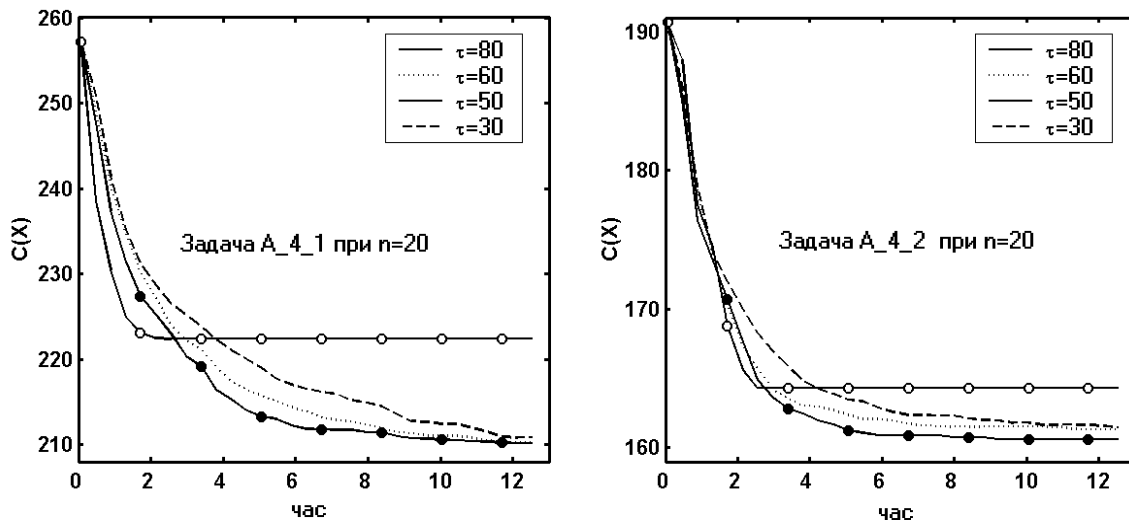


Рис. 5.18. Динаміка генетичної оптимізації з відсічною селекцією

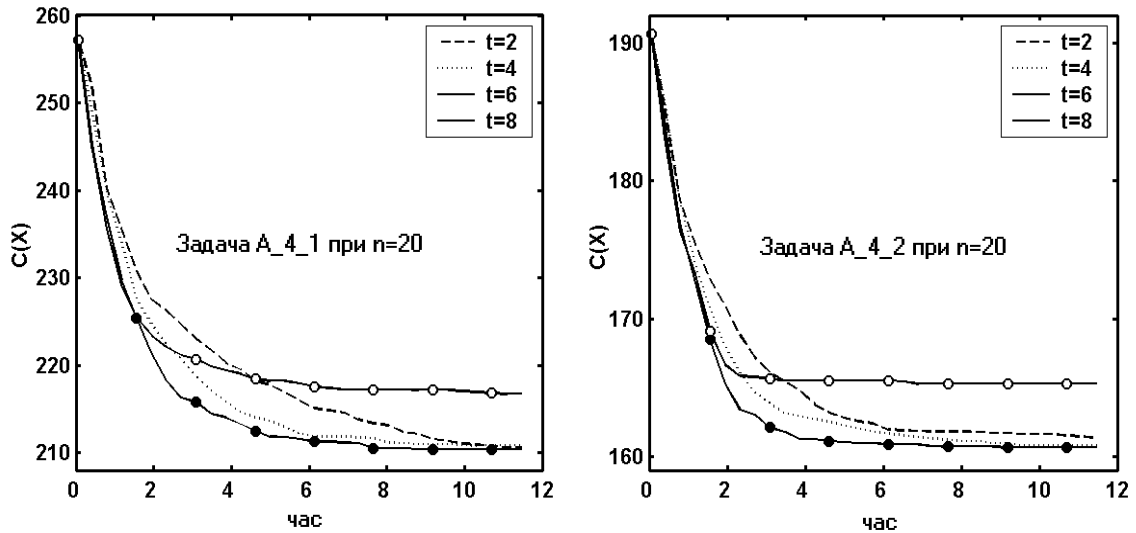


Рис. 5.19. Динаміка генетичної оптимізації з турнірною селекцією

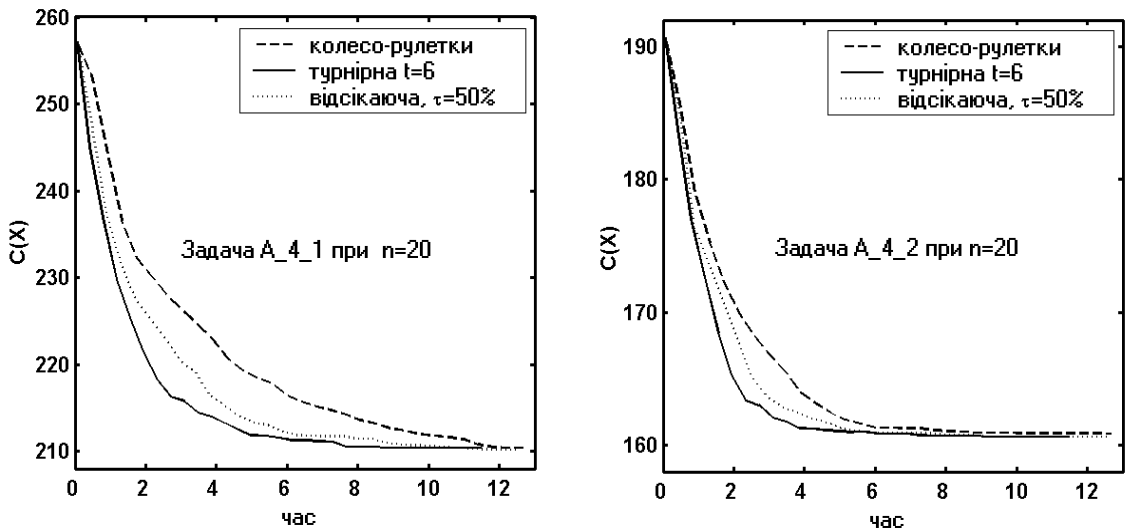


Рис. 5.20. Динаміка генетичної оптимізації з різними стратегіями селекції

Рис. 5.20 свідчить, що селекція за колесом-рулетки має найгіршу динаміку. Динаміки відсічної селекції і турнірної селекції порівнянні, але ГА з турнірною селекцією знаходить оптимум швидше.

### 5.6.3. Генетичний розподіл контрольних точок при обмежених ресурсах

Для оптимізації розподілу контрольних точок використаємо ГА оптимізації з підрозділу 5.1. Операції кодування, схрещування та мутації описані в підрозділах 5.2 та 5.3. Для оцінювання якості розв'язків пропонується використовувати таку фітнес-функцію [109]

$$F(\mathbf{X}) = \begin{cases} p^1(\mathbf{X}), & \text{якщо } \mathbf{X} \text{ допустимий розв'язок} \\ p^1(\mathbf{X}) \cdot (1 - D(\mathbf{X})), & \text{інакше} \end{cases} \quad (5.13)$$

Штрафну функцію  $D(\mathbf{X})$  пропонується визначати таким чином [86, 110]

$$D(\mathbf{X}) = \frac{1}{m+1} \cdot \left( \max \left( 0, \frac{C(\mathbf{X}) - C^*}{C^*} \right) + \sum_{j=1}^m \left( \frac{\Delta_j(\mathbf{X})}{\Delta_j^{\max}} \right)^\alpha \right). \quad (5.14)$$

Зауважимо, що  $D(\mathbf{X}) \leq 1$ , оскільки кожний доданок в (5.14) не перевищує 1.

ГА оптимізації розподілу контрольних точок протестований на задачах В\_4\_1 та В\_4\_2. Результати оптимізації наведені в табл. 5.5.

Таблиця 5.5  
Результати генетичної оптимізації РКГ тестових задач В 4 1 і В 4 2

задача		$C^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
$n=20$	В 4 1	225,80	0,0055	0,0076	0,0078	0,0113	0,9677
	В 4 2	178,11	0,0088	0,0111	0,0125	0,0141	0,9534
$n=40$	В 4 1	436,60	0,0105	0,0142	0,0151	0,0203	0,9399
	В 4 2	317,70	0,0146	0,0185	0,0207	0,0233	0,9228
$n=60$	В 4 1	540,57	0,0132	0,0169	0,0190	0,0203	0,9306
	В 4 2	525,27	0,0185	0,0232	0,0257	0,0294	0,9032
$n=80$	В 4 1	714,15	0,0240	0,0304	0,0347	0,0432	0,8677
	В 4 2	668,94	0,0260	0,0326	0,0368	0,0411	0,8635
$n=100$	В 4 1	974,48	0,0261	0,0332	0,0386	0,0498	0,8522
	В 4 2	828,57	0,0316	0,0393	0,0440	0,0497	0,8354
$n=120$	В 4 1	1138,81	0,0277	0,0355	0,0406	0,0506	0,8455
	В 4 2	1021,17	0,0368	0,0463	0,0517	0,0580	0,8071

Для прискорення роботи генетичної оптимізації розподілу контрольних точок при обмежених ресурсах пропонується процедура ініціалізації популяції, в якій значення генів генеруються за формулою (5.11), процедура швидкого обчислення показників надійності АП та ефективна схема селекції.

В табл. 5.6 порівнюється частота вгадувань оптимальних генів при випадковій та при запропонованій процедурах ініціалізацій популяції для тестових задач В\_4\_1 та В\_4\_2.

Таблиця 5.6

Середня частота вгадувань  $w$  (для 50 експериментів,  $pop\ size=50$ )

Задача		$n=20$	$n=40$	$n=60$	$n=80$	$n=100$	$n=120$	Середня
В_4_1	випадкова ініціалізація	0,49	0,48	0,50	0,49	0,48	0,48	0,48
	запропонована ініціалізація	0,60	0,61	0,59	0,70	0,62	0,66	0,63
В_4_2	випадкова ініціалізація	0,51	0,49	0,50	0,48	0,49	0,50	0,50
	запропонована ініціалізація	0,69	0,72	0,68	0,69	0,65	0,69	0,69

На рис. 5.21–5.23 порівнюються динаміки генетичного розподілу контрольних точок з різними стратегіями селекцій. На рис. 5.21 зображена динаміка оптимізації при різних порогах  $\tau$  відсічної селекції.

Добрі результати досягаються при  $\tau \approx 50\%$ . На рис. 5.22 зображено динаміку оптимізації при різній кількості  $t$  учасників змагань турнірної селекції. Добрі результати досягаються при  $t \approx 6$ .

Рис. 5.23 свідчать, що селекція за колесом-рулетки має найгіршу динаміку. Динаміки відсічної селекції і турнірної селекції порівнянні, але ГА з турнірною селекцією знаходить оптимум швидше.

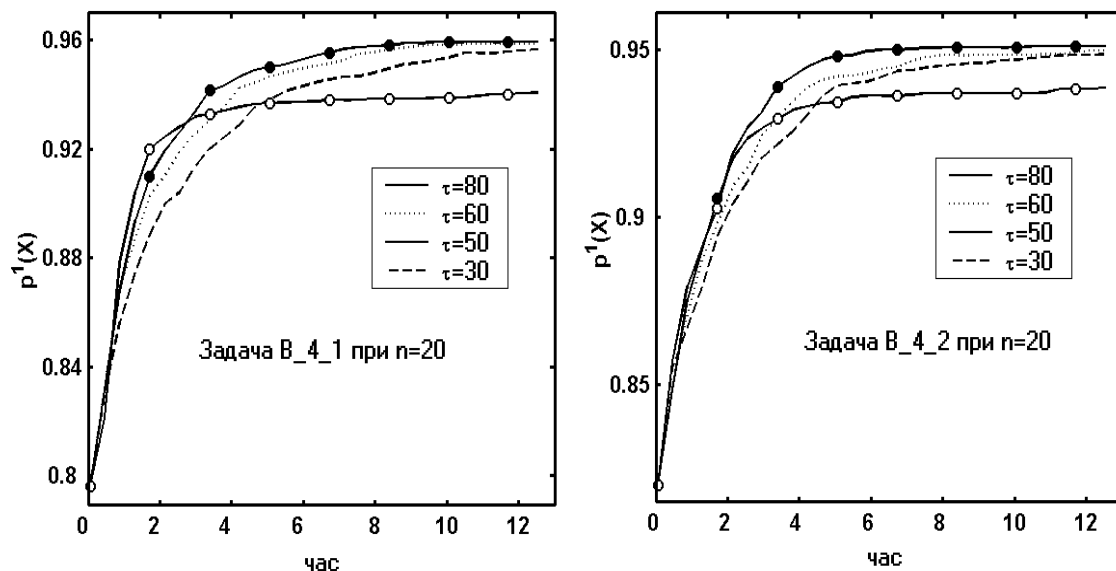


Рис. 5.21. Динаміка генетичної оптимізації з відсічною селекцією

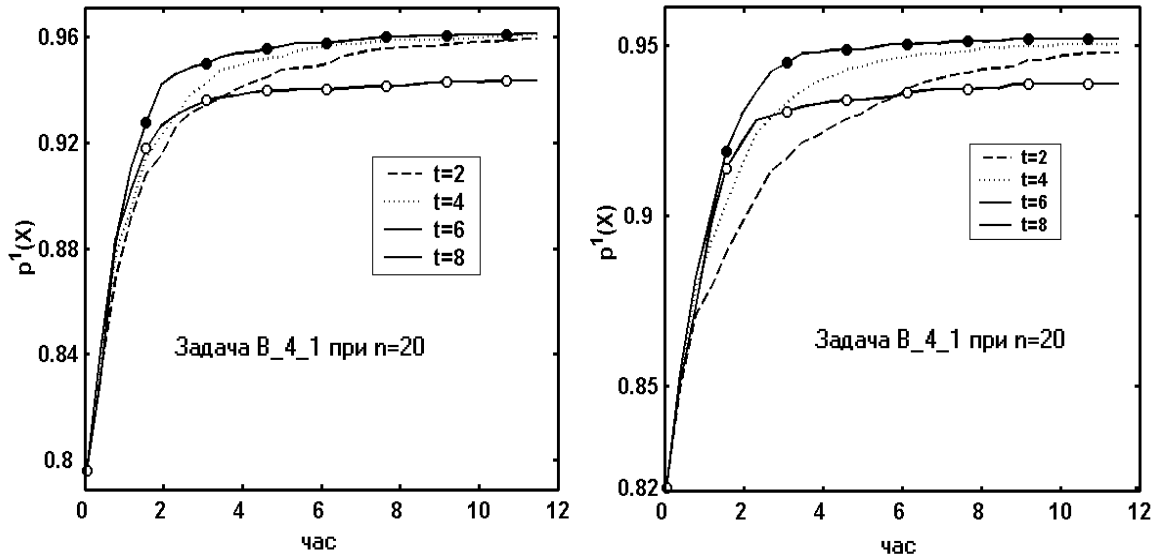


Рис. 5.22. Динаміка генетичної оптимізації з турнірною селекцією

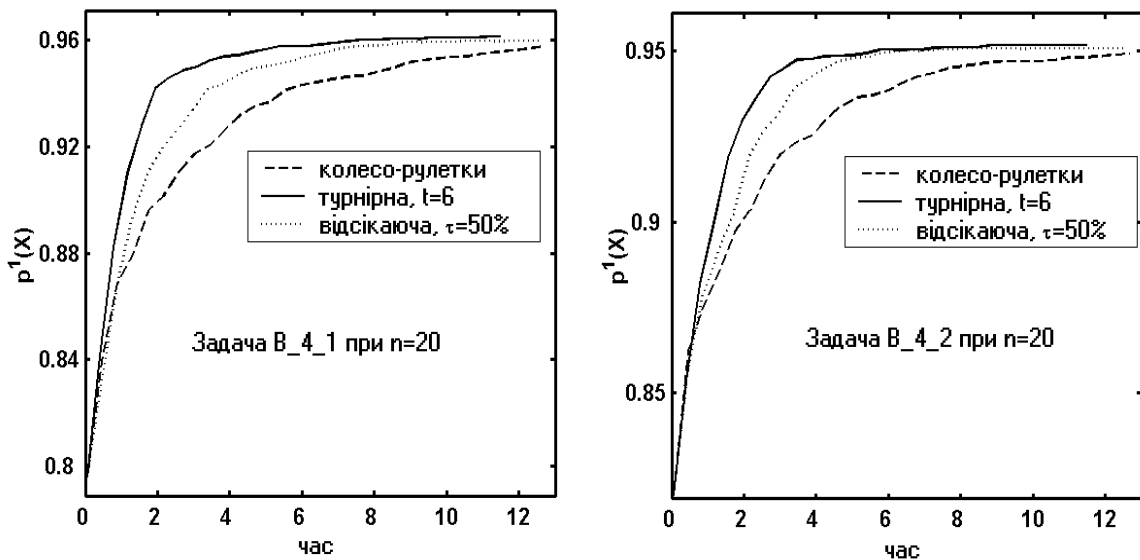


Рис. 5.23. Динаміка генетичної оптимізації з різними стратегіями селекцій

#### 5.6.4. Генетичний вибір кратностей контролів при обмеженнях на безпомилковість

Для оптимізації вибору кратностей контролів будемо використовувати генетичний алгоритм оптимізації з підрозділу 5.1. Операції кодування, схрещування та мутації описані в підрозділах 5.2 та 5.3. Як фітнес-функцію будемо використовувати формулу (5.9). ГА оптимізації кратностей контролів протестований на задачах А\_4\_1 та А\_4\_2. Результати оптимізації наведені в табл. 5.7.

Таблиця 5.7

Результати генетичної оптимізації ВКК тестових задач А\_4\_1 і А\_4\_2

задача		$P^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	C
n=20	А_4_1	0,9637	0,0065	0,0086	0,0092	0,0120	195,67
	А_4_2	0,9569	0,0079	0,0094	0,0105	0,0153	153,27
n=40	А_4_1	0,9451	0,0096	0,0130	0,0139	0,0184	451,97
	А_4_2	0,9272	0,0128	0,0161	0,0182	0,0257	322,35
n=60	А_4_1	0,9306	0,0134	0,0166	0,0188	0,0206	532,88
	А_4_2	0,9010	0,0177	0,0212	0,0258	0,0343	478,45
n=80	А_4_1	0,8681	0,0237	0,0305	0,0344	0,0433	710,40
	А_4_2	0,8727	0,0243	0,0303	0,0344	0,0382	697,51
n=100	А_4_1	0,8531	0,0260	0,0331	0,0386	0,0492	975,04
	А_4_2	0,8451	0,0294	0,0364	0,0413	0,0477	859,29
n=120	А_4_1	0,8383	0,0290	0,0371	0,0421	0,0535	1105,1
	А_4_2	0,8105	0,0356	0,0444	0,0505	0,0590	958,92

Для прискорення роботи генетичної оптимізації вибору кратностей контролів нижче пропонуються процедура ініціалізації популяції, процедура швидкого обчислення показників надійності АП та вибір ефективної схеми селекції.

Процедура ініціалізації початкової популяції генерує значення генів  $x_i$  за такою формулою [109, 110]

$$x_i = \text{ціла\_частина} \left( (1 - \lambda_i) \cdot \underline{x}_i + \lambda_i \cdot \bar{x}_i \right), \quad i = \overline{1, n}, \quad (5.15)$$

де  $\lambda_i = \xi_i^{g(\gamma_i)}$  – число з діапазону  $[0, 1]$ , яке визначає положення  $x_i$  на відріжку  $[\underline{x}_i, \bar{x}_i], i = \overline{1, n}$ ;

$\underline{x}_i$  – нижня межа кратності  $i$ -го контролю,  $i = \overline{1, n}$ ;

$\bar{x}_i$  – верхня межа кратності  $i$ -го контролю,  $i = \overline{1, n}$ .

З пункту 5.7.1 відомо, що  $\lim_{\gamma_i \rightarrow \infty} \lambda_i \rightarrow 1$  та  $\lim_{\gamma_i \rightarrow 0} \lambda_i \rightarrow 0$ . Тому для

ефективних контролів формула (5.15) спрямовує  $x_i$  до  $\bar{x}_i$ , а для неефективних до  $\underline{x}_i$ .

Порівняємо ефективність запропонованої процедури ініціалізації, яка генерує значення генів за формулою (5.15), з випадковою ініціалізацією. Ефективність будемо оцінювати за критерієм (5.12). В табл. 5.8 порівнюється частота вгадувань оптимальних генів випадкової процедури ініціалізації з запропонованою.



Таблиця 5.8

Середня частота вгадувань  $w$  (для 50 експериментів,  $pop\ size=50$ )

Задача		$n=20$	$n=40$	$n=60$	$n=80$	$n=100$	$n=120$	Середня
A_4_1	випадкова ініціалізація	0,38	0,37	0,40	0,41	0,38	0,39	0,38
	запропонована ініціалізація	0,50	0,48	0,50	0,54	0,51	0,49	0,52
A_4_2	випадкова ініціалізація	0,42	0,38	0,39	0,39	0,39	0,40	0,39
	запропонована ініціалізація	0,60	0,47	0,50	0,53	0,54	0,56	0,54

На рис. 5.24 порівнюються розподілення часу генетичної оптимізації запропонованої та випадкової процедури ініціалізації популяції для задачі A\_4\_2 при  $n=20$ . Рис. 5.24 свідчить, що запропонована процедура ініціалізації популяції прискорює генетичну оптимізацію більше ніж в 2 рази.

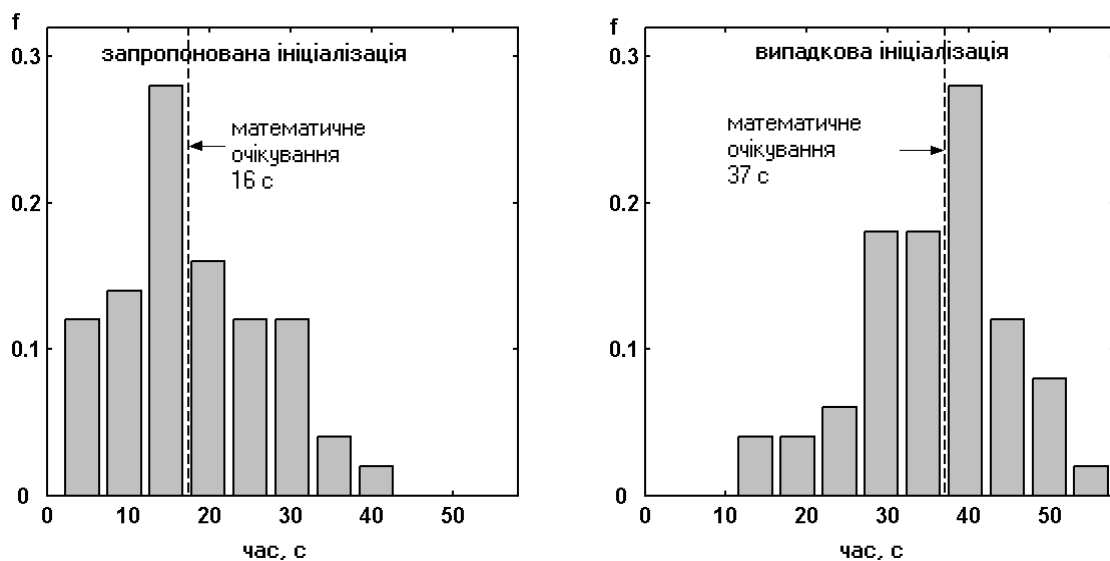


Рис. 5.24. Порівняння запропонованої і випадкової процедур ініціалізації популяції для задачі A\_4\_2 при  $n=20$  (статистика 50 обчислювальних експериментів)

*Процедура швидкого обчислення показників надійності АП.* На рис. 5.25а зображено результат профілювання генетичної оптимізації вибору кратностей контролів для тестової задачі A\_4\_2 при  $n=20$ . З цього рисунка видно, що найбільше часу витрачається на обчислення показників надійності АП за формулами (4.11)–(4.14). На рис. 5.25б зображено результати профілювання генетичного вибору кратностей

контролів з швидким способом обчислення показників надійності АП, який використовувався в задачі розподілу контрольних точок.

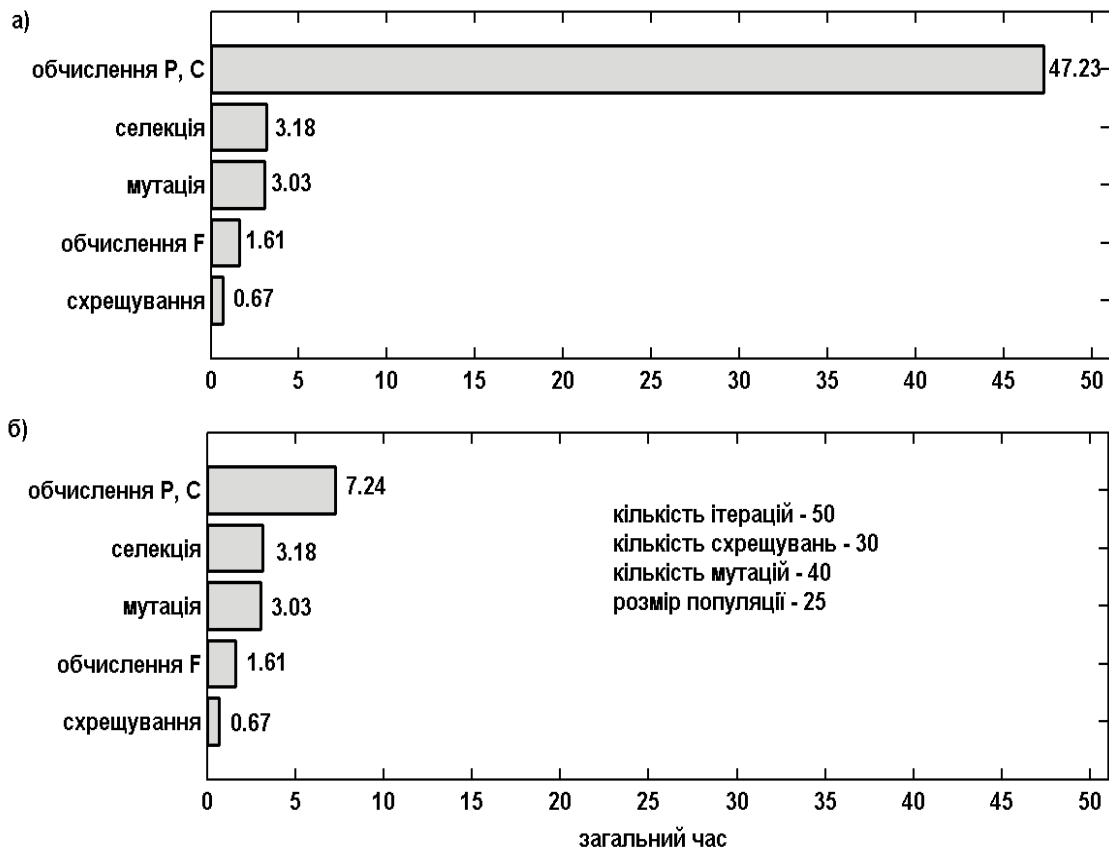


Рис. 5.25. Результати профілювання генетичного ВКК з обчисленням показників надійності АП:  
а) звичайним способом; б) швидким способом

*Вибір ефективної схеми селекції.* На рис. 5.26–5.28 порівнюються середні динаміки генетичного вибору кратностей контролів з різними стратегіями селекцій.

На рис. 5.26 зображено динаміку оптимізації при різних порогах  $\tau$  відсічної селекції. Задовільні результати досягаються при  $\tau \approx 50\%$ . На рис. 5.27 зображено динаміку оптимізації задач при різній кількості  $t$  учасників змагань турнірної селекції. Добрі результати досягаються при  $t \approx 6$ .

Рис. 5.28 свідчить, що селекція за колесом-рулетки має найгіршу динаміку. Динаміки відсічної селекції і турнірної селекції порівнянні, але ГА з турнірною селекцією знаходить оптимум швидше.

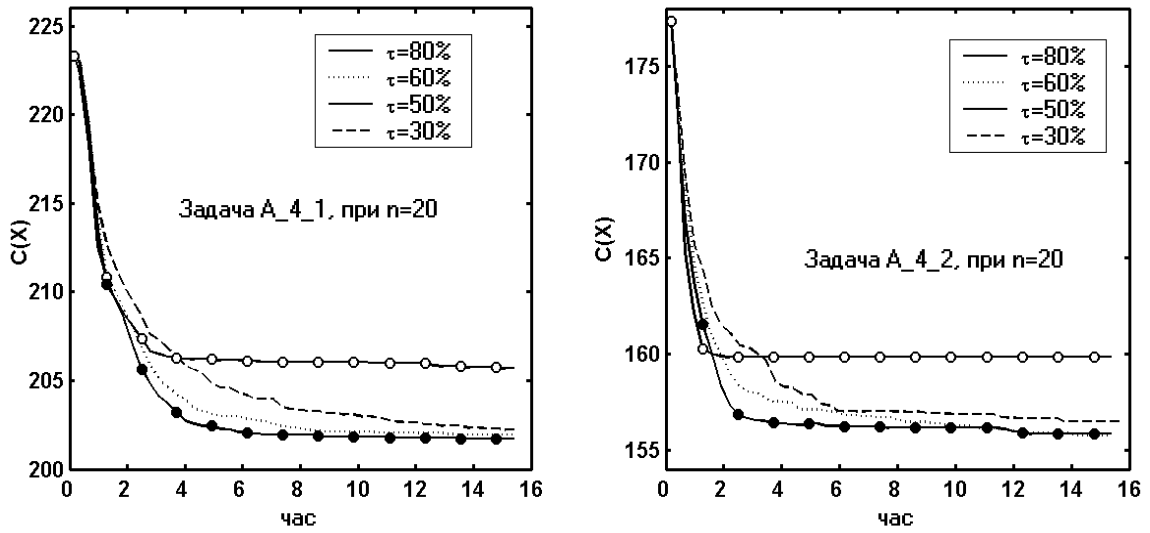


Рис. 5.26. Динаміка генетичної оптимізації з відсічною селекцією

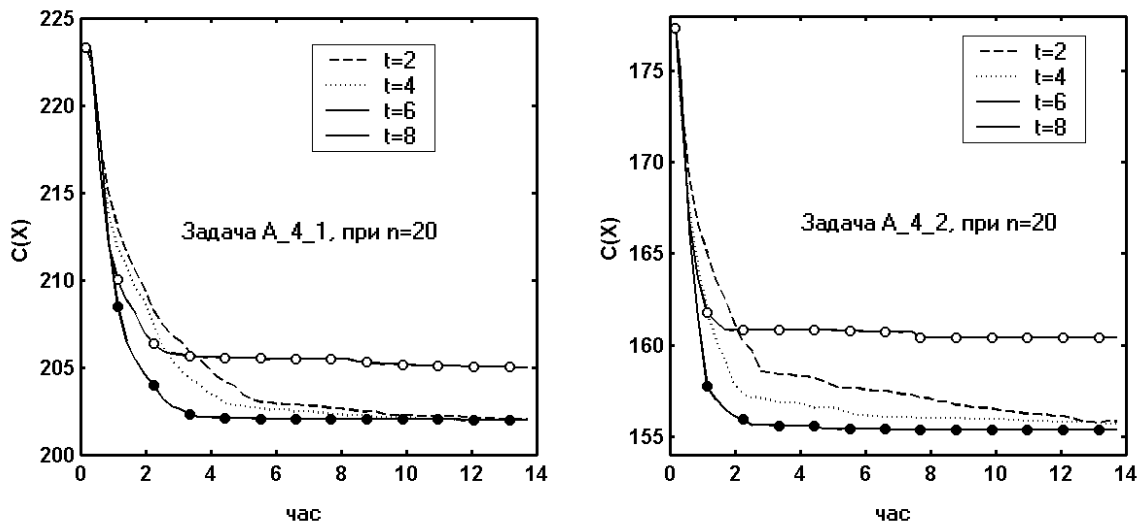


Рис. 5.27. Динаміка генетичної оптимізації з турнірною селекцією

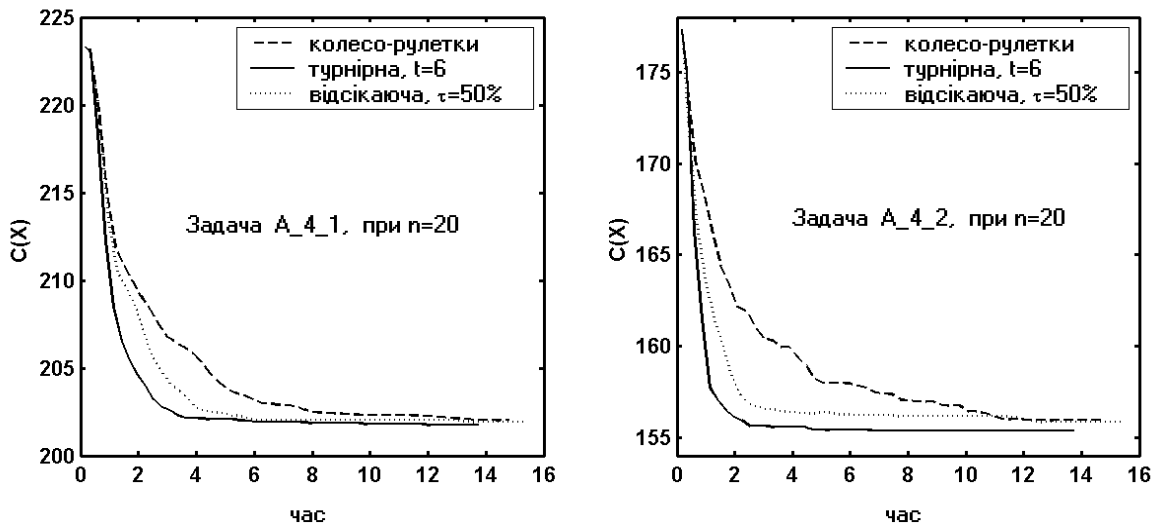


Рис. 5.28. Динаміка генетичної оптимізації з різними стратегіями селекції

### 5.6.5. Генетичний вибір кратностей контролів при обмежених ресурсах

Для оптимізації вибору кратностей контролів будемо використовувати ГА оптимізації з підрозділу 5.1. Операції кодування, схрещування та мутації описані в підрозділах 5.2 та 5.3. Як фітнес-функцію будемо використовувати формулу (5.13). ГА оптимізації вибору кратностей контролів протестований на задачах В\_4\_1 та В\_4\_2. Результати оптимізації наведені в табл. 5.9.

Для прискорення роботи генетичної оптимізації вибору кратностей контролів при обмежених ресурсах пропонуються процедура ініціалізації популяції, в якій значення генів генеруються за формулою (5.15), процедура швидкого обчислення показників надійності АП та ефективна схема селекції.

Таблиця 5.9  
Результати генетичної оптимізації ВКК тестових задач В\_4\_1 і В\_4\_2

задача		$C^*$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
$n=20$	В_4_1	225,21	0,0053	0,0070	0,0074	0,0111	0,9693
	В_4_2	179,67	0,0083	0,0110	0,0122	0,0138	0,9546
$n=40$	В_4_1	440,51	0,0101	0,0136	0,0145	0,0193	0,9425
	В_4_2	317,28	0,0146	0,0183	0,0206	0,0232	0,9233
$n=60$	В_4_1	540,66	0,0133	0,0165	0,0185	0,0208	0,9309
	В_4_2	525,18	0,0183	0,0229	0,0254	0,0294	0,9040
$n=80$	В_4_1	714,78	0,0237	0,0303	0,0336	0,0426	0,8697
	В_4_2	669,45	0,0257	0,0323	0,0363	0,0408	0,8649
$n=100$	В_4_1	979,60	0,0257	0,0326	0,0382	0,0487	0,8548
	В_4_2	829,01	0,0312	0,0389	0,0435	0,0491	0,8372
$n=120$	В_4_1	1140,85	0,0274	0,0350	0,0396	0,0498	0,8482
	В_4_2	1021,05	0,0364	0,0457	0,0514	0,0575	0,8091

В табл. 5.10 порівнюється частота вгадувань оптимальних генів випадкової процедури ініціалізації з ініціалізацією за формулою (5.15) для задач В\_4\_1 та В\_4\_2. Частоту вгадувань оптимальних генів розраховано за формулою (5.12).

На рис. 5.29–5.31 порівнюються середні динаміки генетичного вибору кратностей контролів з різними стратегіями селекцій. На рис. 5.29 зображено динаміку оптимізації при різних порогах  $\tau$  відсічної селекції. Добрі результати досягаються при  $\tau \approx 50\%$ . На рис. 5.30 зображено оптимізацію задач при різній кількості  $t$  учасників змагань турнірної селекції. Добрі результати досягаються при  $t \approx 6$ .

Таблиця 5.10

Середня частота вгадувань  $w$  (для 50 експериментів,  $pop\ size=50$ )

Задача		$n=20$	$n=40$	$n=60$	$n=80$	$n=100$	$n=120$	Середня
В_4_1	випадкова ініціалізація	0,36	0,39	0,40	0,40	0,39	0,39	0,38
	запропонована ініціалізація	0,42	0,49	0,49	0,56	0,50	0,48	0,50
В_4_2	випадкова ініціалізація	0,39	0,40	0,41	0,40	0,39	0,41	0,40
	запропонована ініціалізація	0,51	0,60	0,57	0,56	0,55	0,54	0,56

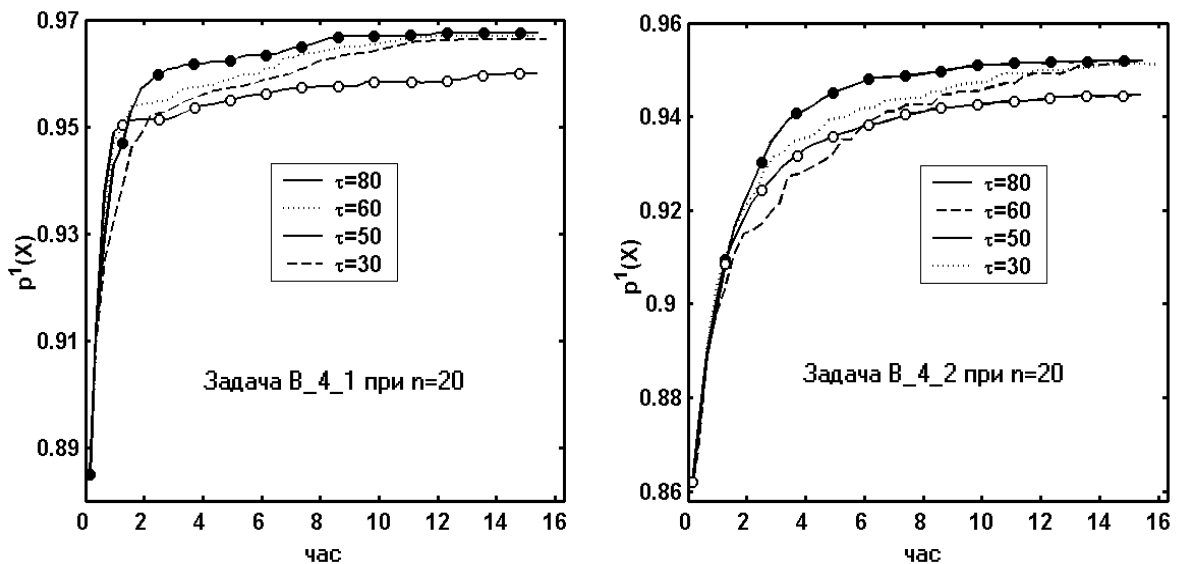


Рис. 5.29. Динаміка генетичної оптимізації з відсичною селекцією

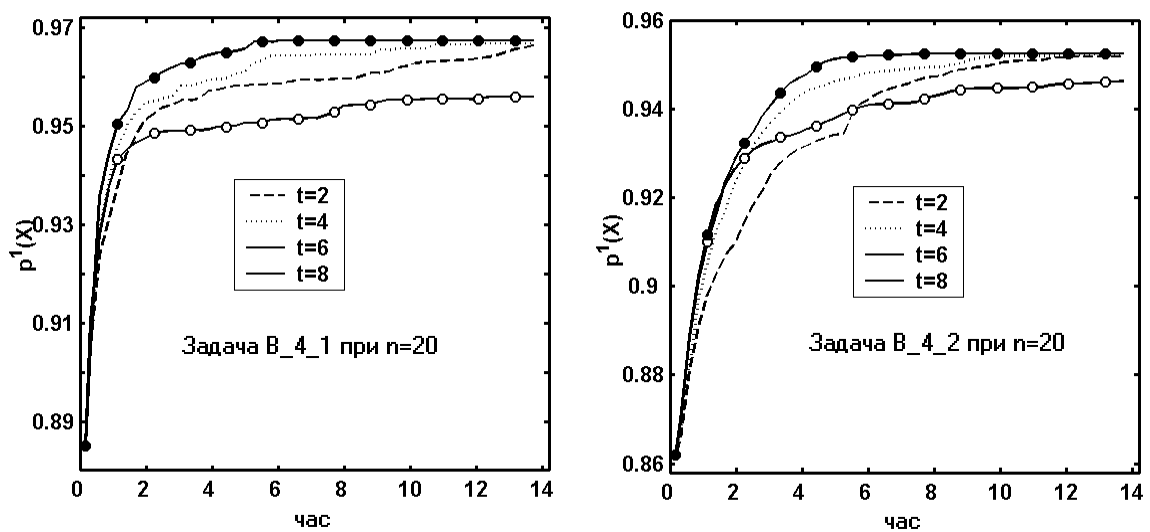


Рис. 5.30. Динаміка генетичної оптимізації з турнірною селекцією

Рис. 5.31 свідчить, що селекція за колесом-рулетки має найгіршу динаміку. Динаміки відсічної селекції і турнірної селекції порівнянні, але ГА з турнірною селекцією знаходить оптимум швидше.

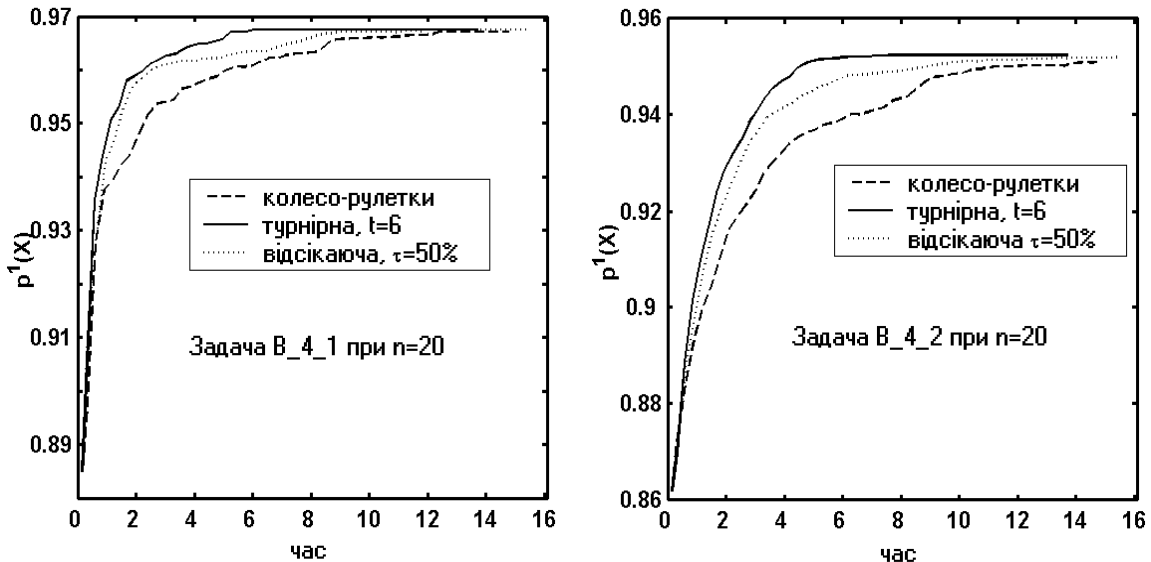


Рис. 5.31. Динаміка генетичної оптимізації з різними стратегіями селекції

### 5.6.6. Порівняння алгоритмів оптимізації надійності багатовимірних алгоритмічних процесів

На рис. 5.32 порівнюються як знайдені за градієнтними та генетичними алгоритмами розв'язки задач В\_4\_1 задовольняють обмеження (4.6) та (4.16). На рис. 5.33 порівнюються розв'язки задач В\_4\_2. З цих рисунків видно, що ГА оптимізації знаходить розв'язки, які задовольняють всі обмеження, тоді як градієнтні не завжди знаходять допустимий розв'язок.

Оцінимо складність градієнтних та ГА оптимізацій, яку будемо визначати як кількість обчислень показників надійності АП та градієнту контролю. Тоді для градієнтних алгоритмів оптимізації складність визначимо так:

$$O_{gp} = O_I + O_{II},$$

де  $O_I$  ( $O_{II}$ ) – складність першої та другої ітеративних ділянок градієнтного алгоритму оптимізації.

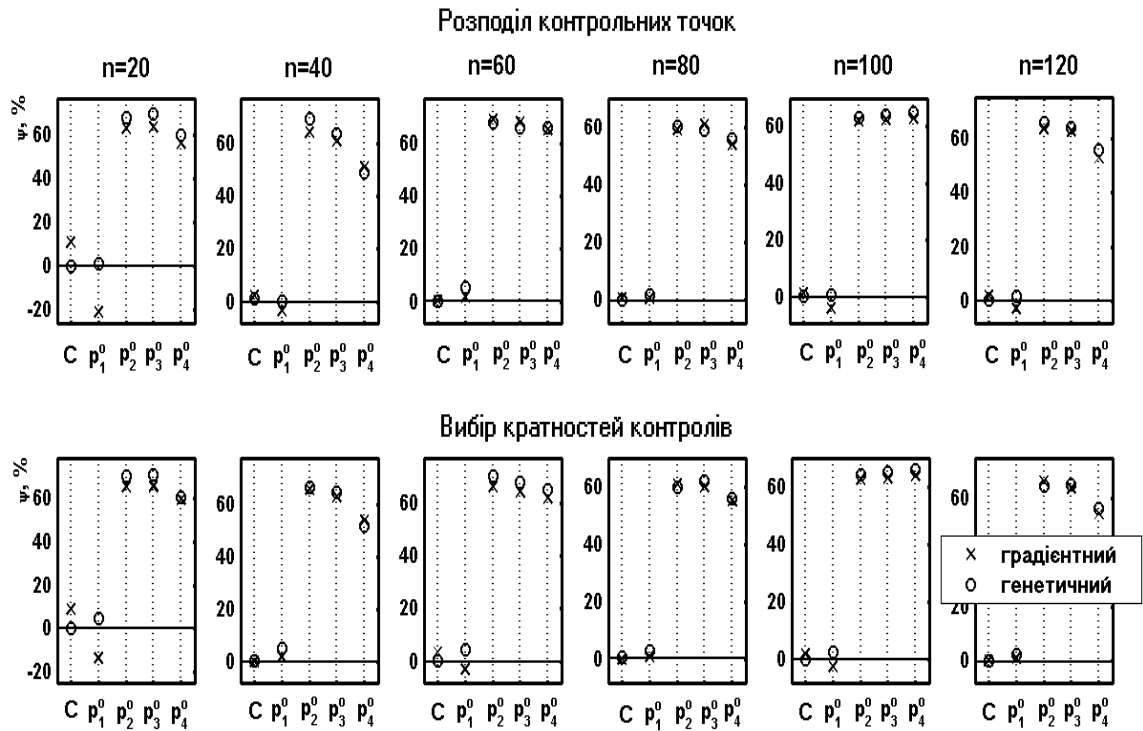


Рис. 5.32. Виконання оптимальними розв'язками обмежень задач В\_4\_1

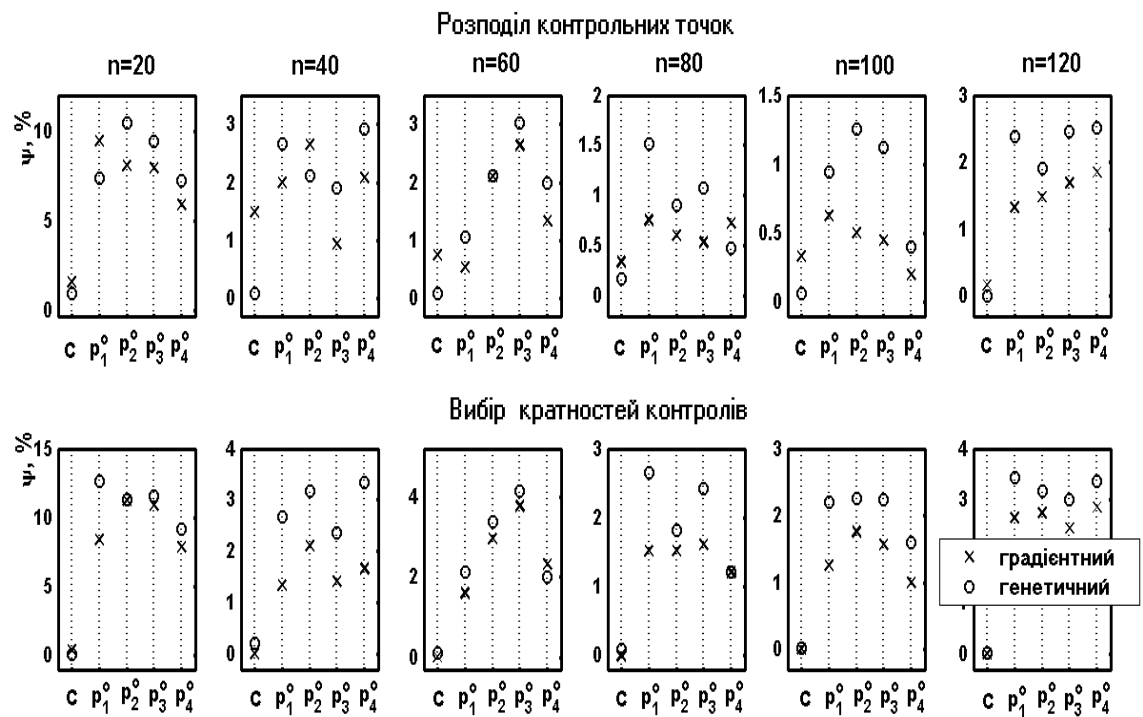


Рис. 5.33. Виконання оптимальними розв'язками обмежень задач В\_4\_2

Визначимо складність розподілу контрольних точок. На першій ділянці цієї задачі оптимізації кількість обчислень градієнта контролю дорівнює  $\sum_{i=1, k_1} (n-i+1) = \frac{k_1}{2} \cdot (2 \cdot n - k_1)$ , де  $k_1$  – кількість встановлених контролів,  $1 \leq k_1 \leq n$ . Кількість обчислень показників надійності АП на цій ділянці дорівнює  $k_1 + 1$ . Тоді  $O_I = \frac{k_1}{2} \cdot (2 \cdot n - k_1) + (k_1 + 1)$ . На другій ділянці обчислюються лише показники надійності АП, кількість яких складає  $Q_{II} = k_2 + \sum_{i=1, k_2} (n-i+1) = k_2 + \frac{k_2}{2} \cdot (2 \cdot n - k_2)$ , де  $k_2$  – кількість надлишкових контролів при прямій постановці задачі або додаткових контролів при зворотній постановці,  $1 \leq k_2 \leq n$ . Тоді складність оптимізації розподілу контрольних точок складає

$$O_{cp} = \frac{k_1}{2} \cdot (2 \cdot n - k_1) + (k_1 + 1) + k_2 + \frac{k_2}{2} \cdot (2 \cdot n - k_2) = (n+1) \cdot (k_1 + k_2) - \frac{1}{2} \cdot (k_1^2 + k_2^2) + 1.$$

Визначимо нижню та верхню межу  $O_{cp}$ . Мінімальна складність досягається при  $k_1 = 1$  та  $k_2 = 0$ . Вона становить  $O_{cp} = n + 2$ . Максимальна, тобто теоретична, складність досягається при  $k_1 = n$  та  $k_2 = n - 1$ . Вона складає

$$O_{cp} = (n+1) \cdot (n+n-1) - \frac{1}{2} \cdot (n^2 + (n-1)^2) + 1 = n^2 + 2n - 1.$$

Таким чином складність алгоритму оптимізації розподілу контрольних точок задовольняє таку умову:

$$n + 2 \leq O_{cp} \leq n^2 + 2n - 1.$$

Для обчислення градієнта контролю та показників надійності за формулами (4.25) та (4.10)–(4.11) необхідно виконати  $m^2 \cdot n$  елементарних операцій. Таким чином, при оптимізації розподілу контрольних точок кількість елементарних операцій задовольняє таку умову

$$m^2 \cdot n \cdot (n+2) \leq O_{cp}^* \leq m^2 \cdot n \cdot (n^2 + 2 \cdot n - 1).$$

Визначимо складність вибору кратностей контролю. На першій ділянці цієї задачі оптимізації складність дорівнює  $O_I = (k_1 + 1) + k_1 \cdot n$ , а на другій  $O_{II} = k_2 \cdot (n + 1)$ , причому  $1 \leq k_1; k_2 \leq n \cdot k; k \geq 1$ . Тоді складність оптимізації вибору кратностей



контролю складає

$$O_{zp} = (k_1 + 1) + k_1 \cdot n + k_2 \cdot (n + 1) = (k_1 + k_2) \cdot (n + 1) + 1.$$

Мінімальна складність –

$$O_{zp} = (k_1 + k_2) \cdot (n + 1) + 1 \geq 2 \cdot (n + 1) + 1 = 2n + 3,$$

а максимальна –

$$O_{zp} = (k_1 + k_2) \cdot (n + 1) + 1 \leq 2n \cdot k \cdot (n + 1) + 1 = 2 \cdot k \cdot n^2 + 2 \cdot k \cdot n + 1.$$

Таким чином складність градієнтної оптимізації вибору кратностей контролів задовольняє умову

$$2n + 3 \leq O_{zp} \leq 2 \cdot k \cdot n^2 + 2 \cdot k \cdot n + 1, \quad k \geq 1.$$

Для обчислення показників надійності за формулами (4.14) та (4.15) необхідно виконати  $m^2 \cdot n$  елементарних операцій. Таким чином, при оптимізації вибору кратностей контролів кількість елементарних операцій задовольняє таку умову:

$$m^2 \cdot n \cdot (2 \cdot n + 3) \leq O_{zp}^* \leq m^2 \cdot n \cdot (2 \cdot k \cdot n^2 + 2 \cdot k \cdot n + 1), \quad k \geq 1.$$

Для ГА оптимізації розподілу контрольних точок та вибору кратностей контролів складність визначимо співвідношенням

$$O_{ген} = O_{ини} + O_{г.оп},$$

де  $O_{ини}$  та  $O_{г.оп}$  – складність процедури ініціалізації та проведення генетичних операцій.

Складність процедури ініціалізації дорівнює  $O_{ини} = n + pop\_size$ , а проведення генетичних операцій  $O_{г.оп} = num\_iter \cdot (2 \cdot num\_chrom + num\_mut)$ . Тоді  $O_{ген}$  запишемо так:

$$O_{ген} = n + pop\_size + num\_iter \cdot (2 \cdot num\_chrom + num\_mut).$$

Оскільки складність обчислення градієнта контролю та показників надійності пропорційні величині  $m^2 \cdot n$ , то останню рівність перепишемо так:

$$O_{ген} = m^2 \cdot n \cdot (n + pop\_size + num\_iter \cdot (2 \cdot num\_chrom + num\_mut)).$$

В табл. 5.11 порівнюються складності градієнтних та ГА оптимізацій для задач розподілу контрольних точок та вибору кратностей контролю.

Таблиця 5.11

Складність градієнтних та генетичних моделей оптимізації РКТ і ВКК

Складність	Градієнтний		Генетичний	
	РКТ	ВКК	РКТ	ВКК
Нижня межа	$m^2 \cdot n \cdot (n + 2)$	$m^2 \cdot n \cdot (2 \cdot n + 3)$	$m^2 \cdot n \cdot (n + pop\_size + num\_iter \cdot (2 \cdot num\_chrom + num\_mut))$	
Верхня межа	$m^2 \cdot n \cdot r,$ $r = (n^2 + 2 \cdot n - 1)$	$m^2 \cdot n \cdot (2 \cdot k \cdot n^2 + 2 \cdot k \cdot n + 1)$ $k \geq 1$		

Генетичний пошук є стохастичним, тому неможливо теоретично визначити, скільки разів потрібно розраховувати цільову функцію, щоб досягти оптимуму. Тому швидкодію визначимо експериментально.

На рис. 5.34 та рис. 5.35 порівнюються час оптимізації генетичного та градієнтного вибору кратностей контролів для задач А\_4\_1, А\_4\_2 та В\_4\_1, В\_4\_2. Оптимізація градієнтними моделями проводилася з використанням швидких процедур обчислення градієнта та розрахунку показників надійності АП. Час оптимізації за генетичними моделями розраховано як середнє значення 100 обчислювальних експериментів. Для порівняння на рисунках наведено швидкодію типового генетичного алгоритму з роботи [86].

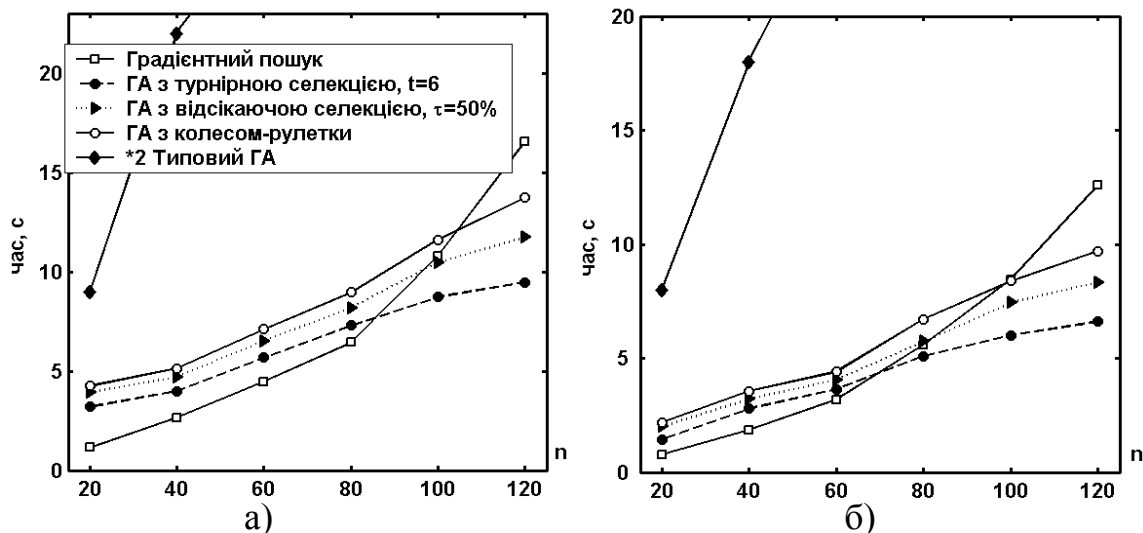


Рис. 5.34. Середній час розв'язання градієнтного та генетичного ВКК з різними схемами селекцій: а) для задач В\_4\_1; б) для задач В\_4\_2

З рис. 5.34 та рис. 5.35 видно, що для задач великої розмірності запропоновані генетичні алгоритми знаходять розв'язки навіть швидше, ніж градієнтні. Причому, найшвидше оптимальний розв'язок знаходять генетичні алгоритми з турнірною селекцією. Таким чином градієнтні алгоритми доцільно використовувати при дефіциті часу, наприклад, для on-line адаптації АП. Генетичні алгоритми корисні на етапі проектування АП.

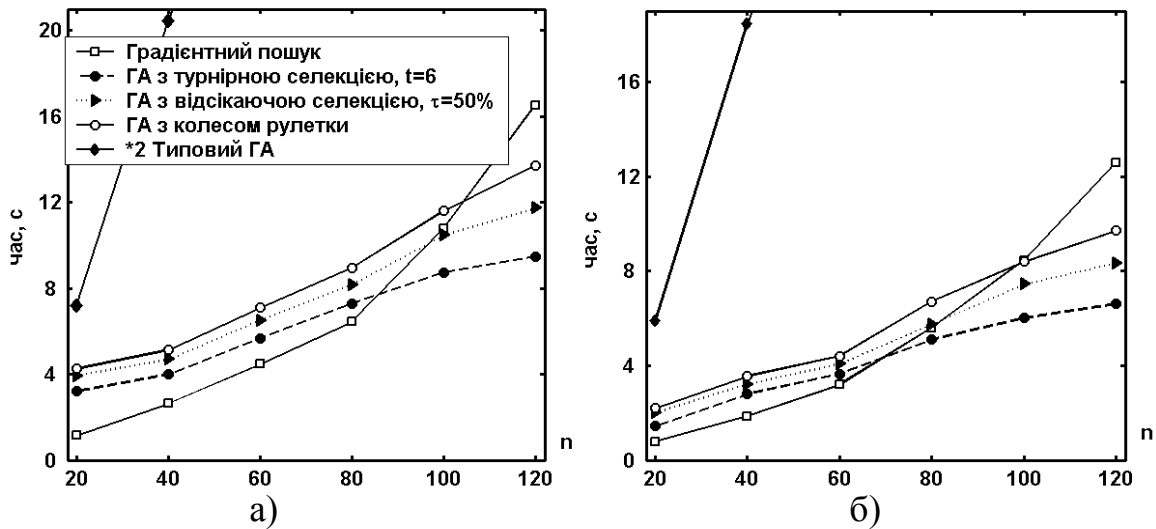


Рис. 5.35. Середній час розв'язання градієнтного та генетичного ВКК з різними схемами селекцій: а) для задач А\_4\_1; б) для задач А\_4\_2

### 5.6.7. Дослідження вірогідності алгоритмів оптимізації надійності багатовимірних алгоритмічних процесів

Розглянемо задачу розподілу контрольних точок, яка використана у проектуванні АП для виготовлення друкованих плат хімічним методом. АП виготовлення друкованої плати за цим методом складається з таких потенційно-контрольованих робочих операторів [75]:

- $A_1$  – отримання заготовок та фіксуючих отворів;
- $A_2$  – підготовка поверхні при отриманні рисунка фотоспособом;
- $A_3$  – отримання захисного рельєфу;
- $A_4$  – травлення міді;
- $A_5$  – видалення захисного рельєфу сіткографією;
- $A_6$  – отримання захисної маски;
- $A_7$  – нанесення знаків маркування;
- $A_8$  – лудіння контактних площадок сплавом Розе;
- $A_9$  – отримання монтажних отворів;
- $A_{10}$  – обробка контуру.

Характеристики бездефектності та трудомісткості відповідних контрольних та доробчих операцій наведені в [75]. Необхідно спроектувати АП виготовлення друкованих плат, щоб при мінімальній трудомісткості  $P^* = 0,80$  (пряма постановка).

Розв'язок цієї задачі отримано в [75] градієнтним методом за бінарною концепцією врахування помилок. Оптимальний вектор контролів  $\mathbf{X} = \{0, 1, 1, 1, 1, 0, 1, 1, 1, 0\}$  забезпечує  $p^1(\mathbf{X}) = 0,808$  та  $C(\mathbf{X}) = 189,3$ .

Для розв'язання цієї задачі застосуємо градієнтний алгоритм оптимізації розподілу контрольних точок з підрозділу 4.4.3. Кількість типів помилок дорівнює  $m=1$ . В табл. 5.12 наведено динаміку оптимізації. З цієї таблиці видно, що результати градієнтної оптимізації розподілу контрольних точок, при якій враховуються помилки різних типів, збігаються з реальним результатом в граничному випадку, коли кількість помилок дорівнює одиниці. Цим підтверджується вірогідність запропонованих градієнтних алгоритмів оптимізації розподілу контрольних точок. ГА з підрозділу 5.7.1 знайшов оптимальний розв'язок в 3 ітерації.

Таблиця 5.12

Динаміка оптимізації РКТ

№	X										$p^1(\mathbf{X})$	$p^0(\mathbf{X})$	$C(\mathbf{X})$
1	0	0	0	0	0	0	0	0	0	0	0,2423	0,7577	115,20
2	0	1	0	0	0	0	0	0	0	0	0,2788	0,7212	115,84
3	0	1	0	0	0	0	1	0	0	0	0,3083	0,6917	116,53
4	0	1	0	0	0	0	1	0	0	1	0,3112	0,6888	117,05
5	0	1	0	0	0	1	1	0	0	1	0,3142	0,6858	117,59
6	0	1	1	0	0	1	1	0	0	1	0,4086	0,5914	132,42
7	0	1	1	0	0	1	1	0	1	1	0,4727	0,5273	143,34
8	0	1	1	0	1	1	1	0	1	1	0,5718	0,4282	158,01
9	0	1	1	1	1	1	1	0	1	1	0,6848	0,3152	173,29
10	0	1	1	1	1	1	1	1	1	1	0,8243	0,1757	190,31
11	0	1	1	1	1	1	1	1	1	0	0,8165	0,1835	189,79
12	0	1	1	1	1	0	1	1	1	0	0,8088	0,1912	189,25

Розглянемо задачу вибору кратностей контролів, яка використовується в [76] для мінімізації витрат на контроль АП. Процес складається з 4-х потенційно контрольованих робочих операцій. Ймовірність правильного виконання АП має бути не менша за  $P^* > 0,9995$ . Початкові дані наведено в [76]. Розв'язком цієї задачі є вектор  $\mathbf{X} = (4, 4, 4, 3)$

кратностей контролів, який забезпечує  $p^1(\mathbf{X}) = 0,99952$  та  $C(\mathbf{X}) = 69,19$ .

Для розв'язання цієї задачі застосуємо з підрозділу 4.4.5 градієнтний алгоритм оптимізації вибору кратностей контролів при  $m=1$ . Координати початкової точки оптимізації такі [76]:  $\mathbf{X} = (4, 2, 3, 1)$ . Динаміка оптимізації кратностей контролю наведена в табл. 5.13. З цієї таблиці видно, що результати градієнтної оптимізації вибору кратностей контролю, при якій враховуються помилки різних типів, збігаються з результатом в [76] для граничного випадку, коли кількість типів помилок дорівнює 1. ГА з підрозділу 5.6.4 знайшов оптимальний розв'язок за чотири ітерації.

Таблиця 5.13

Динаміка оптимізації ВКК

№	X				$p^1(\mathbf{X})$	$p^0(\mathbf{X})$	$C(\mathbf{X})$
0	4	2	3	1	0,9976	0,0024	47,10
1	4	2	3	2	0,9983	0,0017	50,10
2	4	2	4	2	0,9987	0,0013	54,19
3	4	2	4	3	0,9988	0,0012	57,19
4	5	2	4	3	0,99897	0,00103	62,19
5	5	3	4	3	0,99947	0,00053	68,18
6	5	4	4	3	0,99966	0,00034	74,23
7	4	4	4	3	0,99952	0,00048	69,19

Порівняємо розв'язки градієнтних та ГА оптимізацій, які наведені в підрозділах 4.4.3–4.4.6 та 5.6.1–5.6.4, з глобальними розв'язками за методом повного перебору. Для тестових задач при  $n=20$ , повним перебором отримані глобальні розв'язки задач розподілу контрольних точок та вибору кратностей контролів. Ці розв'язки збігаються з генетичними. Градієнтні розв'язки є локальними, про що свідчить табл. 5.14. В цій таблиці наведено розбіжність градієнтних розв'язків з глобальними. Час повного перебору для задачі розподілу контрольних точок склав 15 хвилин, а для задачі вибору кратностей контролів – 37 хвилин. Нагадаємо, що на розв'язання цих задач генетичним пошуком знадобилось 2 с та 4 с відповідно.

Порівняємо градієнтні алгоритми оптимізації надійності, які запропоновані в підрозділі 4.4 з бінарними алгоритмами оптимізації, які запропоновані в [79]. Для порівняння треба перерахувати характеристики надійності з багатовимірної концепції врахування помилок в бінарну. Такий перерахунок пропонується здійснити таким чином:

$$k_{\omega_i}^{00} = \frac{1}{m} \cdot \sum_{j=1}^m k_{\omega_i}^{00j} \quad \text{та} \quad v_{U_i}^1 = \frac{1}{m} \cdot \sum_{j=1}^m v_{U_i}^{1j}, \quad i = \overline{1, n}.$$

Показники надійності АП, які синтезовані за бінарним градієнтним алгоритмом оптимізації наведені в табл. 5.15. Значення цих показників отримані підстановкою оптимальних векторів контролю в формули (4.4), (4.5) та (4.13), (4.14).

Таблиця 5.14

Розбіжність градієнтних та глобальних розв'язків для тестових задач, в яких  $n=20$  ( $гл$  – глобальний розв'язок,  $гр$  – градієнтний розв'язок)

Задача		$C_{гл} - C_{гр}$	$p_{гл}^{0_1} - p_{гр}^{0_1}$	$p_{гл}^{0_2} - p_{гр}^{0_2}$	$p_{гл}^{0_3} - p_{гр}^{0_3}$	$p_{гл}^{0_4} - p_{гр}^{0_4}$	$p_{гл}^1 - p_{гр}^1$
РКТ	A 4 1	-20.5	-0,0001	-0,0001	-0,0004	0,0005	0,0001
	A 4 2	-1	0,0002	0,0001	0,0002	0,0002	-0,0007
	B 4 1	24.3	-0,0012	-0,0012	-0,0016	-0,0012	0,0051
	B 4 2	1.8	0,0002	-0,0003	-0,0001	-0,0002	0,0004
ВКК	A 4 1	-7.3	-0,0001	-0,0001	-0,0001	-0,0003	0,0005
	A 4 2	-1	0,0002	0,0001	0,0002	0,0003	-0,0008
	B 4 1	20	-0,0010	-0,0012	-0,0014	-0,0004	0,0040
	B 4 2	0.5	-0,0004	-0,0000	-0,0001	-0,0002	0,0007

Таблиця 5.15

Показники надійності АП, які синтезовані за градієнтною бінарною моделлю оптимізації (жирний шрифт вказує недопустимі розв'язки)

Розподіл контрольних точок	$N$	множина	$C$	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
	20	A 4 1	226,52	0,0065	0,0087	0,0092	0,0122	0,9635
A 4 2		157,28	0,0077	0,0093	0,0104	0,0151	0,9575	
B 4 1		219,17	<b>0,0060</b>	0,0080	0,0085	0,0108	0,9667	
B 4 2		179,53	0,0092	0,0123	0,0137	0,0151	0,9498	
40	A 4 1	457,7	0,0095	0,0130	0,0136	0,0182	0,9457	
	A 4 2	328,78	0,0129	0,0158	0,0185	0,0255	0,9273	
	B 4 1	438,01	<b>0,0106</b>	0,0142	0,0153	0,0198	0,9401	
	B 4 2	313,21	0,0148	0,0184	0,0208	0,0235	0,9225	
60	A 4 1	544,95	0,0134	0,0164	0,0184	0,0206	0,9312	
	A 4 2	488,43	0,0175	0,0216	0,0248	0,0349	0,9012	
	B 4 1	538,44	0,0137	0,0167	0,0188	0,0210	0,9299	
	B 4 2	524,74	<b>0,0188</b>	0,0233	0,0257	0,0298	0,9025	

Продовження таблиці 5.15

Розподіл контрольних точок	N	множина	C	$p^{0_1}$	$p^{0_2}$	$p^{0_3}$	$p^{0_4}$	$p^1$
	80	A 4 1	736,41	<b>0,0240</b>	0,0307	0,0338	0,0430	0,8686
		A 4 2	711,78	0,0243	0,0303	0,0345	0,0382	0,8727
		B 4 1	712,38	<b>0,0247</b>	0,0317	0,0350	0,0436	0,8650
		B 4 2	669,68	<b>0,0265</b>	0,0329	0,0369	<b>0,0414</b>	0,8623
	100	A 4 1	985,54	0,0260	0,0327	0,0379	0,0501	0,8534
		A 4 2	867,92	0,0293	0,0364	0,0411	<b>0,0479</b>	0,8453
		B 4 1	977,71	<b>0,0265</b>	0,0334	0,0386	0,0509	0,8507
		B 4 2	828,65	<b>0,0320</b>	0,0397	0,0445	<b>0,0504</b>	0,8337
120	A 4 1	1121,20	<b>0,0297</b>	0,0381	0,0424	0,0541	0,8358	
	A 4 2	971,46	0,0356	0,0442	0,0500	0,0593	0,8109	
	B 4 1	1138,23	<b>0,0286</b>	0,0365	0,0408	0,0517	0,8425	
	B 4 2	1019,70	<b>0,0374</b>	<b>0,0467</b>	0,0518	<b>0,0588</b>	0,8054	

В табл. 5.16 наведено кількість недопустимих розв'язків задач з бібліотеки [5], які отримані при бінарній та багатовимірній оптимізаціях надійності АП.

Таблиця 5.16

Порівняння кількості недопустимих розв'язків різними алгоритмами оптимізації

Задача	A 4 1		A 4 2		B 4 1		B 4 2		Всього	
	РКТ	ВКК	РКТ	ВКК	РКТ	ВКК	РКТ	ВКК	РКТ	ВКК
Багатовимірне	0	0	0	0	4	3	0	0	4	3
Бінарне	2	3	0	2	5	4	4	1	16	11

З табл. 5.16 видно, що з 24 задач розподілу контрольних точок за бінарним градієнтним алгоритмом знайдено 16 недопустимих розв'язків, а за запропованою – 4. З 24 задач вибору кратностей контролів за бінарним градієнтним алгоритмом знайдено 11 недопустимих розв'язків, а за запропованою – 3. Таким чином запроповані багатовимірні градієнтні алгоритми значно краще за бінарні алгоритми розв'язують задачі оптимізації надійності АП з урахуванням помилок різних типів. Крім того, зменшення багатовимірними градієнтними алгоритмами кількості недопустимих розв'язків підвищує якість проектування АП.

## 6. НЕЧІТКІ МОДЕЛІ НАДІЙНОСТІ БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

Узагальнюються багатовимірні моделі надійності операторів, логічних умов і типових алгоритмічних структур на випадок нечітких початкових даних. Застосування моделей ілюструється на прикладі прогнозування надійності АП проведення платіжного доручення в інтернетівській системі «клієнт-банк». В розділі використовуються матеріали робіт [82, 87, 89].

### 6.1. Нечіткі початкові дані

Невизначені ймовірісно-часові характеристики надійності операторів та логічних умов алгоритмів будемо задавати в формі (1.1). В нечіткій теорії надійності для формалізації лінгвістичних термів зазвичай використовують трикутну, трапецієподібну, гаусову та дзвіноподібну функції належності. Аналітичні вирази та графіки цих функцій належностей наведено в табл. 6.1 та на рис. 6.1.

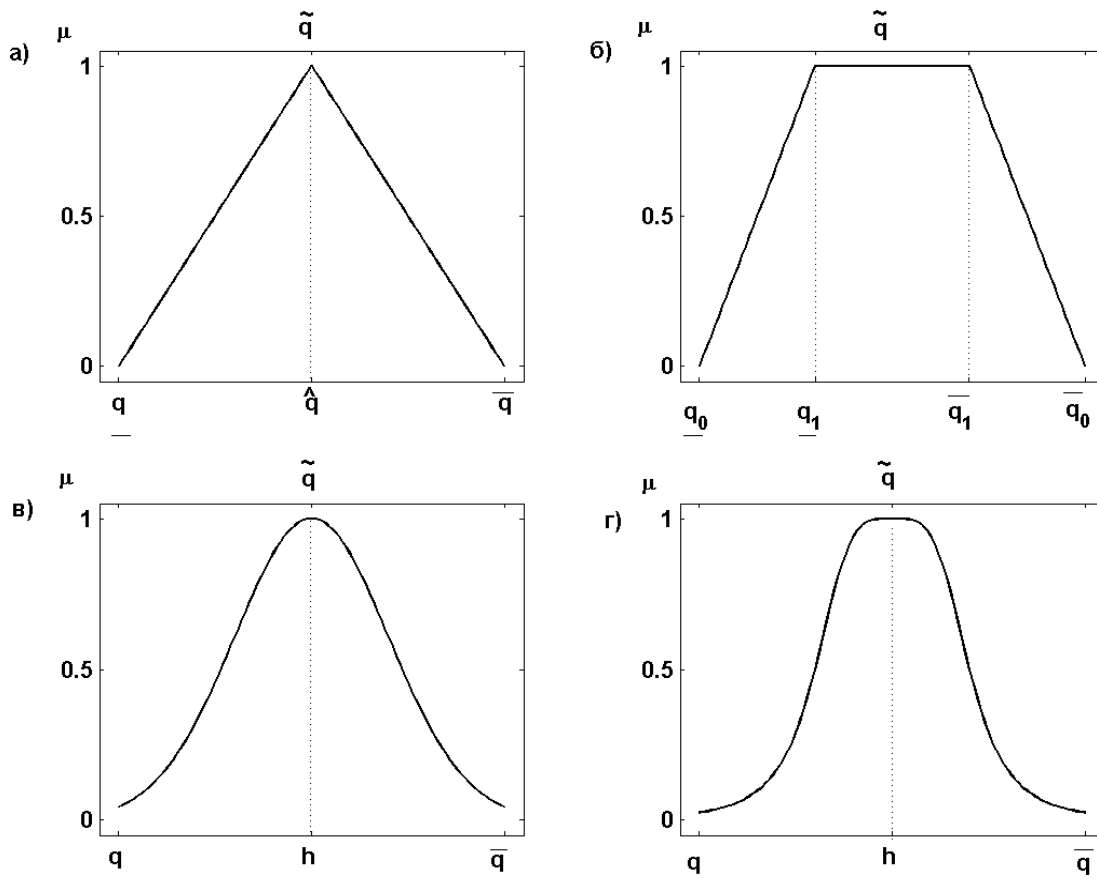


Рис. 6.1. Графіки функцій належності:  
а) трикутна; б) трапецієподібна; в) гаусова; г) дзвіноподібна



Таблиця 6.1

## Моделі функцій належностей [82, 135]

Назва	Аналітичний вираз	Перехід до $\alpha$ -форми
Трикутна	$\mu_{\tilde{q}}(q) = \begin{cases} 0, & \text{якщо } q < \underline{q} \\ \frac{q - \underline{q}}{\hat{q} - \underline{q}}, & \text{якщо } \underline{q} \leq q < \hat{q} \\ \frac{\hat{q} - q}{\hat{q} - \bar{q}}, & \text{якщо } \hat{q} \leq q \leq \bar{q} \\ 0, & \text{якщо } q > \bar{q} \end{cases},$ <p>де <math>\hat{q}</math> – координата максимуму;  <math>(\underline{q}, \bar{q})</math> – носій нечіткого числа <math>\tilde{q}</math></p>	$\underline{q}_\alpha = \underline{q} + (\hat{q} - \underline{q}) \cdot \alpha,$ $\bar{q}_\alpha = \bar{q} - (\bar{q} - \hat{q}) \cdot \alpha$
Трапеціє-подібна	$\mu_{\tilde{q}}(q) = \begin{cases} 0, & \text{якщо } q < \underline{q}_0 \\ \frac{q - \underline{q}_0}{\underline{q}_1 - \underline{q}_0}, & \text{якщо } \underline{q}_0 \leq q < \underline{q}_1 \\ 1, & \text{якщо } \underline{q}_1 \leq q < \bar{q}_1 \\ \frac{\bar{q}_1 - q}{\bar{q}_1 - \bar{q}_0}, & \text{якщо } \bar{q}_1 \leq q \leq \bar{q}_0 \\ 0, & \text{якщо } q > \bar{q}_0 \end{cases}$ <p>де <math>(\underline{q}_0, \bar{q}_0)</math> – носій числа <math>\tilde{q}</math>;  <math>[\underline{q}_1, \bar{q}_1]</math> – ядро нечіткого числа <math>\tilde{q}</math></p>	$\underline{q}_\alpha = \underline{q}_0 + (\underline{q}_1 - \underline{q}_0) \cdot \alpha$ $\bar{q}_\alpha = \bar{q}_0 - (\bar{q}_0 - \bar{q}_1) \cdot \alpha$
Гаусова	$\mu_{\tilde{q}}(q) = \exp\left(-\frac{(q - h)^2}{2 \cdot c^2}\right),$ <p>де <math>h</math> – координата максимуму;  <math>c</math> – коефіцієнт концентрації</p>	$\underline{q}_\alpha = h - c \sqrt{2 \cdot \ln\left(\frac{1}{\alpha}\right)},$ $\bar{q}_\alpha = h + c \sqrt{2 \cdot \ln\left(\frac{1}{\alpha}\right)}$
Дзвіно-подібна	$\mu_{\tilde{q}}(q) = \frac{1}{1 + \left \frac{q - h}{c}\right ^{2b}},$ <p>де <math>b</math> – коефіцієнт крутизни</p>	$\underline{q}_\alpha = c - h \cdot \left(\frac{1 - \alpha}{\alpha}\right)^{1/2b}$ $\bar{q}_\alpha = c + h \cdot \left(\frac{1 - \alpha}{\alpha}\right)^{1/2b}$

Виконувати розрахунки з нечіткими числами зручно, якщо представити їх в  $\alpha$ -формі (1.1). Перехід від трикутної, трапецієподібної, гаусівської та дзвіноподібної функцій належностей до  $\alpha$ -форми здійснюється за формулами з табл. 6.1.

Вплив факторів на лінгвістичні значення показників надійності можна врахувати за допомогою нечіткої бази знань. Нечітка база знань це сукупність таких логічних висловлювань [55, 72, 74, 82]

$$\bigcup_{p=1}^{s_j} \left[ \bigcap_{i=1}^n (x_i = a_i^{jp}) \right] \rightarrow q = l_j, \quad j = \overline{1, k}, \quad (6.3)$$

де  $\cup$  ( $\cap$ ) – логічні операції «ТА» («АБО»);

$l_j$  ( $j = \overline{1, k}$ ) – лінгвістична оцінка параметра  $q$ , яка визначається на терм-множині  $\{l_1, l_2, \dots, l_k\}$ ;

$a_i^{jp}$  – лінгвістична оцінка фактора  $x_i$  в правилі з номером  $jp$  ( $i = \overline{1, n}, j = \overline{1, k}, p = \overline{1, s_j}$ );

$s_j$  ( $j = \overline{1, k}$ ) – кількість правил, яка визначає значення параметра  $q = l_j$ .

Ідентифікація залежностей «входи – вихід» на основі нечітких баз знань розглянута в [72]. Застосування нечітких баз знань для ідентифікації залежностей надійності при моделюванні АП розглянуто в роботах [82, 83].

## 6.2. Нечіткі моделі надійності операторів та логічних умов

Нечіткі характеристики надійності робочого оператора  $A$  задамо такою матрицею

$$\tilde{\mathbf{P}}_A = \begin{pmatrix} \tilde{p}_A^1 & \tilde{p}_A^{0_1} & \dots & \tilde{p}_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}, \quad (6.4)$$

де  $\tilde{p}_A^1$  – нечітка ймовірність правильного виконання оператора  $A$ ;

$\tilde{p}_A^{0_j}$  – нечітка ймовірність внесення помилки  $j$ -го типу при виконанні оператора  $A$ , причому  $\tilde{p}_A^1 + \sum_{j=1}^m \tilde{p}_A^{0_j} = \tilde{1}$ ,  $j = \overline{1, m}$ , де  $\tilde{1}$  – «нечітка

одиниця» [82]:  $\tilde{1} = \bigcup_{\alpha \in [0,1]} (e_\alpha, \bar{e}_\alpha)$ ,  $e_\alpha \leq 1$ ;  $\bar{e}_\alpha \geq 1$ .

Модель (6.4) узагальнює на нечіткий випадок матричного представлення надійності робочого оператора при  $n$ -арній концепції врахування дефектів.

Нечіткі характеристики надійності оператора доробки  $U$  задамо такою матрицею

$$\tilde{\mathbf{P}}_U = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \tilde{v}_U^{1_1} & \tilde{v}_U^{0_1} & \dots & 0 \\ \dots & & & \\ \tilde{v}_U^{1_m} & 0 & \dots & \tilde{v}_U^{0_m} \end{pmatrix}, \quad (6.5)$$

де  $\tilde{v}_U^{1_j}$  ( $\tilde{v}_U^{0_j}$ ) – нечітка ймовірність усунення (не усунення) помилки  $j$ -го типу при доробці  $U$ , причому  $\tilde{v}_U^{1_j} + \tilde{v}_U^{0_j} = \tilde{1}$ ,  $j = \overline{1, m}$ .

Час виконання робочого оператора  $A$  та оператора доробки  $U$  позначимо нечіткими числами  $\tilde{t}_A$  та  $\tilde{t}_U$ .

Нечіткі характеристики надійності логічної умови  $\omega$  задамо такими матрицями:

$$\tilde{\mathbf{K}}_\omega^1 = \begin{pmatrix} \tilde{k}_\omega^{11} & 0 & \dots & 0 \\ 0 & \tilde{k}_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \tilde{k}_\omega^{01_m} \end{pmatrix}; \quad \tilde{\mathbf{K}}_\omega^0 = \begin{pmatrix} \tilde{k}_\omega^{10} & 0 & \dots & 0 \\ 0 & \tilde{k}_\omega^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \tilde{k}_\omega^{00_m} \end{pmatrix}, \quad (6.6)$$

де  $\tilde{k}_\omega^{11}$  ( $\tilde{k}_\omega^{10}$ ) – нечітка ймовірність того, що відсутність помилок ідентифіковано правильно (неправильно);

$\tilde{k}_\omega^{01_j}$  ( $\tilde{k}_\omega^{00_j}$ ) – нечітка ймовірність пропуску (виявлення) помилки  $j$ -го типу, причому  $\tilde{k}_\omega^{01_j} + \tilde{k}_\omega^{00_j} = \tilde{1}$ ,  $j = \overline{1, m}$ .

Час виконання контролю  $\omega$  позначимо нечітким числом  $\tilde{t}_\omega$ .

### 6.3. Принцип нечіткого узагальнення

Для отримання моделей надійності АП при нечітких початкових даних пропонується використовувати принцип узагальнення. Він формулюється таким чином:

*Принцип узагальнення.* Якщо  $y = f(q_1, q_2, \dots, q_N)$  – функція від  $N$  незалежних аргументів  $q_i, i = \overline{1, N}$ , які задані нечіткими числами

$\tilde{q}_i = \bigcup_{\alpha \in [0, 1]} (q_{i\alpha}, \bar{q}_{i\alpha}), i = \overline{1, N}$ , то значенням функції

$\tilde{y} = f(\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_N)$  називається число  $\tilde{y} = \bigcup_{\alpha \in [0, 1]} (\underline{y}_\alpha, \bar{y}_\alpha)$ , де

$$\underline{y}_\alpha = \inf_{\substack{q_i^* \in [q_{i\alpha}, \bar{q}_{i\alpha}] \\ i=1, N}} f(q_1^*, q_2^*, \dots, q_N^*); \quad (6.7)$$

$$\bar{y}_\alpha = \sup_{\substack{q_i^* \in [q_{i\alpha}, \bar{q}_{i\alpha}] \\ i=1, N}} f(q_1^*, q_2^*, \dots, q_N^*). \quad (6.8)$$

Застосування принципу узагальнення зводиться до розв'язання для кожного  $\alpha$ -зрізу такої задачі оптимізації: знайти максимальне та мінімальне значення функції  $y = f(q_1, q_2, \dots, q_N)$  при умові, що аргументи можуть набувати значень з відповідних  $\alpha$ -зрізів [82]. Кількість  $\alpha$ -зрізів вибирається так, щоб забезпечити необхідну точність обчислень.

Нижню (верхню) межу  $\underline{y}_\alpha$  ( $\bar{y}_\alpha$ ) можна визначити за такою методикою [82]:

1. Представити початкову математичну модель у вигляді функції  $y = f(q_1, q_2, \dots, q_N)$ .

2. Визначити границі зміни аргументів  $q_i \in [q_i, \bar{q}_i], i = \overline{1, N}$ .

3. Знайти частинні похідні  $\frac{\partial y}{\partial q_i}, i = \overline{1, N}$ .

4. Позначити:

$q_r$  – аргументи, для яких  $\frac{\partial y}{\partial q_r} \geq 0$  на всій області визначення;

$q_s$  – аргументи, для яких  $\frac{\partial y}{\partial q_s} \leq 0$  на всій області визначення;

$q_l$  – аргументи, для яких  $\frac{\partial y}{\partial q_l}$  є знакозмінною функцією і її

знак залежить лише від значень аргументів  $q_r$  та  $q_s$ , тобто

$$\frac{\partial y}{\partial q_l} = g_l(q_r, q_s).$$

5. Записати нечітку математичну модель  $\tilde{y} = f(\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_N)$  у вигляді

$$y = \bigcup_{\alpha \in [0,1]} \left\{ f\left(\underline{q}_{r_\alpha}, \bar{q}_{s_\alpha}, q_{l_\alpha}^I\right), f\left(\bar{q}_{r_\alpha}, \underline{q}_{s_\alpha}, q_{l_\alpha}^{II}\right) \right\},$$

$$\text{де } q_{l_\alpha}^I = \begin{cases} \underline{q}_{l_\alpha}, & \text{при } g_1(\underline{q}_r, \bar{q}_s) \geq 0 \\ \bar{q}_{l_\alpha}, & \text{при } g_1(\underline{q}_r, \bar{q}_s) < 0 \end{cases}; \quad q_{l_\alpha}^{II} = \begin{cases} \bar{q}_{l_\alpha}, & \text{при } g_1(\underline{q}_r, \bar{q}_s) \geq 0 \\ \underline{q}_{l_\alpha}, & \text{при } g_1(\underline{q}_r, \bar{q}_s) < 0 \end{cases}.$$

#### 6.4. Нечіткі моделі надійності типових алгоритмічних структур

Моделювання надійності АП будемо проводити з використанням типових алгоритмічних структур: «послідовна», « $\omega$ -диз'юнкція», « $\omega$ -ітерація», «робота-контроль-добробка», «багаторазова робота» та « $\omega$ -добробка».

*Твердження 6.1.* При нечітких ймовірнісно-часових характеристиках операторів алгоритмічна структура «послідовна»  $B = A_1 \cdot A_2$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_B^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{B_\alpha}^1, \bar{p}_{B_\alpha}^1 \right), \quad (6.9)$$

$$\text{де } \underline{p}_B^1 = \underline{p}_{A_1}^1 \cdot \underline{p}_{A_2}^1; \quad \bar{p}_B^1 = \bar{p}_{A_1}^1 \cdot \bar{p}_{A_2}^1;$$

$$\tilde{p}_B^{0j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{B_\alpha}^{0j}, \bar{p}_{B_\alpha}^{0j} \right), \quad j = \overline{1, m}, \quad (6.10)$$

$$\text{де } \underline{p}_B^{0j} = \underline{p}_{A_1}^1 \cdot \underline{p}_{A_2}^{0j} + \underline{p}_{A_1}^{0j}; \quad \bar{p}_B^{0j} = \bar{p}_{A_1}^1 \cdot \bar{p}_{A_2}^{0j} + \bar{p}_{A_1}^{0j};$$

$$\tilde{t}_B = \bigcup_{\alpha \in [0,1]} \left( \underline{t}_{B_\alpha}, \bar{t}_{B_\alpha} \right), \quad (6.11)$$

$$\text{де } \underline{t}_B = \underline{t}_{A_1} + \underline{t}_{A_2}; \quad \bar{t}_B = \bar{t}_{A_1} + \bar{t}_{A_2}.$$

*Доведення.* Початкова математична модель ймовірнісних характеристик структури «послідовна» є такою [79]

$$\mathbf{P}_B = \mathbf{P}_{A_1} \cdot \mathbf{P}_{A_2}. \quad (6.12)$$

Перепишемо вираз (6.12) таким чином

$$\mathbf{P}_B = \begin{pmatrix} p_{A_1}^1 & p_{A_1}^0 & \dots & p_{A_1}^0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{A_2}^1 & p_{A_2}^0 & \dots & p_{A_2}^0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} p_{A_1}^1 \cdot p_{A_2}^1 & p_{A_1}^1 \cdot p_{A_2}^0 + p_{A_1}^0 & \dots & p_{A_1}^0 \cdot p_{A_2}^0 + p_{A_1}^0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

З останньої матриці запишемо ймовірнісні характеристики структури «послідовна»:

$$p_B^1 = p_{A_1}^1 \cdot p_{A_2}^1; \quad (6.13)$$

$$p_B^{0_j} = p_{A_1}^1 \cdot p_{A_2}^{0_j} + p_{A_1}^0, \quad j = \overline{1, m}. \quad (6.14)$$

Початкова модель середнього часу виконання структури «послідовна» визначається за такою формулою [79]

$$t_B = t_{A_1} + t_{A_2}. \quad (6.15)$$

Для узагальнення моделей (6.13)–(6.15) на випадок нечітких початкових даних в табл. 6.2 представлені частинні похідні та їх знаки.

За знаками частинних похідних показників надійності (табл. 6.2) згідно з кроком 5 методики узагальнення отримаємо нечіткі моделі надійності (6.9)–(6.11).

Таблиця 6.2

Знаки частинних похідних показників надійності  
структури «послідовна»

Показник	Частинна похідна	Знак похідної
$p_B^1$	$\frac{\partial p_B^1}{\partial p_{A_1}^1} = p_{A_2}^1; \quad \frac{\partial p_B^1}{\partial p_{A_2}^1} = p_{A_1}^1$	«+»
$p_B^{0_j}, j = \overline{1, m}$	$\frac{\partial p_B^{0_j}}{\partial p_{A_1}^0} = 1; \quad \frac{\partial p_B^{0_j}}{\partial p_{A_2}^0} = p_{A_1}^1$	«+»
$t_B$	$\frac{\partial t_B}{\partial t_{A_1}} = 1; \quad \frac{\partial t_B}{\partial t_{A_2}} = 1$	«+»

*Твердження 6.2.* При нечітких ймовірнісно-часових характеристиках операторів та логічної умови алгоритмічна структура « $\omega$ -диз'юнкція»  $C = (A_1 \vee A_2)$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_C^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{C_\alpha}^1, \overline{p}_{C_\alpha}^1 \right), \quad (6.16)$$

$$\text{де } \underline{p}_C^1 = p_\omega^I \cdot (\underline{p}_{A_2}^1 + k_\omega^I \cdot (\underline{p}_{A_1}^1 - \underline{p}_{A_2}^1));$$

$$\overline{p}_C^1 = \min(1, p_\omega^{II} \cdot (\overline{p}_{A_2}^1 + k_\omega^{II} \cdot (\overline{p}_{A_1}^1 - \overline{p}_{A_2}^1));$$

$$p_\omega^I = \begin{cases} \underline{p}_\omega^1, \underline{p}_{A_1}^1 - \underline{p}_{A_2}^1 \geq 0 \\ \overline{p}_\omega^1, \underline{p}_{A_1}^1 - \underline{p}_{A_2}^1 < 0 \end{cases}; \quad p_\omega^{II} = \begin{cases} \overline{p}_\omega^1, \overline{p}_{A_1}^1 - \overline{p}_{A_2}^1 \geq 0 \\ \underline{p}_\omega^1, \overline{p}_{A_1}^1 - \overline{p}_{A_2}^1 < 0 \end{cases};$$

$$k_\omega^I = \begin{cases} \underline{k}_\omega^{11}, \underline{p}_{A_1}^1 - \underline{p}_{A_2}^1 \geq 0 \\ \overline{k}_\omega^{11}, \underline{p}_{A_1}^1 - \underline{p}_{A_2}^1 < 0 \end{cases}; \quad k_\omega^{II} = \begin{cases} \overline{k}_\omega^{11}, \overline{p}_{A_1}^1 - \overline{p}_{A_2}^1 \geq 0 \\ \underline{k}_\omega^{11}, \overline{p}_{A_1}^1 - \overline{p}_{A_2}^1 < 0 \end{cases};$$

$$\tilde{p}_C^{0_j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{C_\alpha}^{0_j}, \overline{p}_{C_\alpha}^{0_j} \right), \quad j = \overline{1, m}, \quad (6.17)$$

$$\text{де } \underline{p}_C^{0_j} = p_\omega^I \cdot (\underline{p}_{A_1}^{0_j} + k_\omega^I \cdot (\underline{p}_{A_1}^{0_j} - \underline{p}_{A_2}^{0_j})) + \underline{p}_\omega^{0_j};$$

$$\overline{p}_C^{0_j} = p_\omega^{II} \cdot (\overline{p}_{A_1}^{0_j} + k_\omega^{II} \cdot (\overline{p}_{A_1}^{0_j} - \overline{p}_{A_2}^{0_j})) + \overline{p}_\omega^{0_j};$$

$$p_\omega^I = \begin{cases} \underline{p}_\omega^{0_j}, \underline{p}_{A_1}^{0_j} - \underline{p}_{A_2}^{0_j} \geq 0 \\ \overline{p}_\omega^{0_j}, \underline{p}_{A_1}^{0_j} - \underline{p}_{A_2}^{0_j} < 0 \end{cases}; \quad p_\omega^{II} = \begin{cases} \overline{p}_\omega^{0_j}, \overline{p}_{A_1}^{0_j} - \overline{p}_{A_2}^{0_j} \geq 0 \\ \underline{p}_\omega^{0_j}, \overline{p}_{A_1}^{0_j} - \overline{p}_{A_2}^{0_j} < 0 \end{cases};$$

$$k_\omega^I = \begin{cases} \underline{k}_\omega^{11}, \underline{p}_{A_1}^{0_j} - \underline{p}_{A_2}^{0_j} \geq 0 \\ \overline{k}_\omega^{11}, \underline{p}_{A_1}^{0_j} - \underline{p}_{A_2}^{0_j} < 0 \end{cases}; \quad k_\omega^{II} = \begin{cases} \overline{k}_\omega^{11}, \overline{p}_{A_1}^{0_j} - \overline{p}_{A_2}^{0_j} \geq 0 \\ \underline{k}_\omega^{11}, \overline{p}_{A_1}^{0_j} - \overline{p}_{A_2}^{0_j} < 0 \end{cases};$$

$$\tilde{t}_C = \bigcup_{\alpha \in [0,1]} \left( \underline{t}_{C_\alpha}, \overline{t}_{C_\alpha} \right), \quad (6.18)$$

$$\text{де } \underline{t}_C = \underline{t}_\omega + [1 - b_1] \cdot \underline{t}_{A_1} + b_2 \cdot \underline{t}_{A_2}; \quad \bar{t}_C = \bar{t}_\omega + [1 - b_1] \cdot \bar{t}_{A_1} + b_2 \cdot \bar{t}_{A_2},$$

$$b_1 = p_1 \cdot (1 - k_1) + \sum_{j=1}^m p_\omega^{0j} \cdot k_\omega^{0j}; \quad b_2 = p_2 \cdot (1 - k_2) + \sum_{j=1}^m p_\omega^{0j} \cdot k_\omega^{0j};$$

$$p_1 = \begin{cases} \underline{p}_\omega^1, \underline{t}_{A_2} - \underline{t}_{A_1} \geq 0 \\ -\underline{p}_\omega^1, \underline{t}_{A_2} - \underline{t}_{A_1} < 0 \end{cases}; \quad p_2 = \begin{cases} \bar{p}_\omega^1, \bar{t}_{A_2} - \bar{t}_{A_1} < 0 \\ -\bar{p}_\omega^1, \bar{t}_{A_2} - \bar{t}_{A_1} \geq 0 \end{cases};$$

$$k_1 = \begin{cases} \underline{k}_\omega^{11}, \underline{t}_{A_1} - \underline{t}_{A_2} \geq 0 \\ -\underline{k}_\omega^{11}, \underline{t}_{A_1} - \underline{t}_{A_2} < 0 \end{cases}; \quad k_2 = \begin{cases} \bar{k}_\omega^{11}, \bar{t}_{A_1} - \bar{t}_{A_2} < 0 \\ -\bar{k}_\omega^{11}, \bar{t}_{A_1} - \bar{t}_{A_2} \geq 0 \end{cases};$$

$$p_\omega^{0j} = \begin{cases} \underline{p}_\omega^{0j}, \underline{t}_{A_2} - \underline{t}_{A_1} \geq 0 \\ -\underline{p}_\omega^{0j}, \underline{t}_{A_2} - \underline{t}_{A_1} < 0 \end{cases}; \quad p_\omega^{0j} = \begin{cases} \bar{p}_\omega^{0j}, \bar{t}_{A_2} - \bar{t}_{A_1} < 0 \\ -\bar{p}_\omega^{0j}, \bar{t}_{A_2} - \bar{t}_{A_1} \geq 0 \end{cases};$$

$$k_\omega^{0j} = \begin{cases} \underline{k}_\omega^{00j}, \underline{t}_{A_2} - \underline{t}_{A_1} \geq 0 \\ -\underline{k}_\omega^{00j}, \underline{t}_{A_2} - \underline{t}_{A_1} < 0 \end{cases}; \quad k_\omega^{0j} = \begin{cases} \bar{k}_\omega^{00j}, \bar{t}_{A_2} - \bar{t}_{A_1} \geq 0 \\ -\bar{k}_\omega^{00j}, \bar{t}_{A_2} - \bar{t}_{A_1} < 0 \end{cases}.$$

*Доведення.* Початкова математична модель ймовірнісних характеристик структури « $\omega$ -диз'юнкція» задається такою формулою [79]

$$\mathbf{P}_C = \mathbf{P}_\omega^1 \cdot (\mathbf{K}_\omega^1 \cdot \mathbf{P}_{A_1} + \mathbf{K}_\omega^0 \cdot \mathbf{P}_{A_2}). \quad (6.19)$$

Перепишемо вираз (6.19) таким чином

$$\begin{aligned} \mathbf{P}_C &= \mathbf{P}_\omega^1 \cdot \left[ \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01m} \end{pmatrix} \cdot \begin{pmatrix} p_{A_1}^1 & p_{A_1}^{01} & \dots & p_{A_1}^{0m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} + \begin{pmatrix} k_\omega^{10} & 0 & \dots & 0 \\ 0 & k_\omega^{001} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{00m} \end{pmatrix} \cdot \begin{pmatrix} p_{A_2}^1 & p_{A_2}^{01} & \dots & p_{A_2}^{0m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \right] = \\ &= \begin{pmatrix} p_\omega^1 & p_\omega^{01} & \dots & p_\omega^{0m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \left[ \begin{pmatrix} p_{A_1}^1 \cdot k_\omega^{11} & p_{A_1}^{01} \cdot k_\omega^{11} & \dots & p_{A_1}^{0m} \cdot k_\omega^{11} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} + \begin{pmatrix} p_{A_2}^1 \cdot k_\omega^{10} & p_{A_2}^{01} \cdot k_\omega^{10} & \dots & p_{A_2}^{0m} \cdot k_\omega^{10} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \right] = \end{aligned}$$



$$= \begin{pmatrix} p_{\omega}^1 \cdot p_b & p_{\omega}^1 \cdot (p_{A_1}^0 \cdot k_{\omega}^{11} + p_{A_2}^0 \cdot k_{\omega}^{10}) + p_{\omega}^0 & \dots & p_{\omega}^1 \cdot (p_{A_1}^0 \cdot k_{\omega}^{11} + p_{A_2}^0 \cdot k_{\omega}^{10}) + p_{\omega}^0 \\ 0 & 1 & \dots & 0 \\ \ddots & \ddots & \dots & \ddots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

де  $p_b = p_{A_1}^1 \cdot k_{\omega}^{11} + p_{A_2}^1 \cdot k_{\omega}^{10}$ .

З останньої матриці запишемо ймовірнісні характеристики структури « $\omega$ -диз'юнкція»:

$$p_C^1 = p_{\omega}^1 \cdot (p_{A_2}^1 + k_{\omega}^{11} \cdot (p_{A_1}^1 - p_{A_2}^1)); \quad (6.20)$$

$$p_C^{0j} = p_{\omega}^1 \cdot (p_{A_1}^{0j} + k_{\omega}^{11} \cdot (p_{A_1}^{0j} - p_{A_2}^{0j})) + p_{\omega}^0, \quad j = \overline{1, m}. \quad (6.21)$$

Початкова модель середнього часу виконання структури « $\omega$ -диз'юнкція» визначається за такою формулою [79]

$$t_C = t_{\omega} + (1 - b) \cdot t_{A_1} + b \cdot t_{A_2}, \quad (6.22)$$

де  $b = p_{\omega}^1 \cdot k_{\omega}^{10} + \sum_{j=1}^m p_{\omega}^{0j} \cdot k_{\omega}^{00j}$ .

Для узагальнення моделей (6.20)–(6.22) на випадок нечітких початкових даних в табл. 6.3 представлені частинні похідні та їх знаки.

Таблиця 6.3

Знаки частинних похідних показників надійності  
структури « $\omega$ -диз'юнкція»

Показник	Частинна похідна	Знак похідної
$p_C^1$	$\frac{\partial p_C^1}{\partial p_{\omega}^1} = p_{A_2}^1 + k_{\omega}^{11} \cdot (p_{A_1}^1 - p_{A_2}^1)$	$\left\{ \begin{array}{l} \langle\langle + \rangle\rangle, \text{ якщо } (p_{A_1}^1 - p_{A_2}^1) \geq 0 \\ \langle\langle - \rangle\rangle, \text{ якщо } (p_{A_1}^1 - p_{A_2}^1) < 0 \end{array} \right.$
	$\frac{\partial p_C^1}{\partial p_{A_1}^1} = k_{\omega}^{11} \cdot p_{\omega}^1; \quad \frac{\partial p_C^1}{\partial p_{A_2}^1} = p_{\omega}^1 - k_{\omega}^{11} \cdot p_{\omega}^1$	$\langle\langle + \rangle\rangle$
	$\frac{\partial p_C^1}{\partial k_{\omega}^{11}} = p_{\omega}^1 \cdot (p_{A_1}^1 - p_{A_2}^1)$	$\left\{ \begin{array}{l} \langle\langle + \rangle\rangle, \text{ якщо } (p_{A_1}^1 - p_{A_2}^1) \geq 0 \\ \langle\langle - \rangle\rangle, \text{ якщо } (p_{A_1}^1 - p_{A_2}^1) < 0 \end{array} \right.$

Продовження таблиці 6.3

Показник	Частинна похідна	Знак похідної
$p_C^0, j=1, m$	$\frac{\partial p_C^0}{\partial p_\omega^1} = p_{A_1}^0 + k_\omega^{11} \cdot (p_{A_1}^0 - p_{A_2}^0)$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (p_{A_1}^0 - p_{A_2}^0) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (p_{A_1}^0 - p_{A_2}^0) < 0 \end{cases}$
	$\frac{\partial p_C^0}{\partial p_\omega^0} = 1$	$\langle\langle + \rangle\rangle$
	$\frac{\partial p_C^0}{\partial k_\omega^{11}} = p_\omega^1 \cdot (p_{A_1}^0 - p_{A_2}^0)$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (p_{A_1}^0 - p_{A_2}^0) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (p_{A_1}^0 - p_{A_2}^0) < 0 \end{cases}$
	$\frac{\partial p_C^0}{\partial p_{A_1}^0} = p_\omega^1 + p_\omega^1 \cdot k_\omega^{11}$	$\langle\langle + \rangle\rangle$
	$\frac{\partial p_C^0}{\partial p_{A_2}^0} = -p_\omega^1 \cdot k_\omega^{11}$	$\langle\langle - \rangle\rangle$
$t_C$	$\frac{\partial t_C}{\partial t_\omega} = 1 \quad \frac{\partial t_C}{\partial t_{A_2}} = p_\omega^1 \cdot (1 - k_\omega^{11}) + \sum_{j=1, m} p_\omega^0 \cdot k_\omega^{00j}$	$\langle\langle + \rangle\rangle$
	$\frac{\partial t_C}{\partial t_{A_1}} = p_\omega^1 \cdot k_\omega^{11} + \sum_{j=1, m} p_\omega^0 \cdot (1 - k_\omega^{00j})$	$\langle\langle + \rangle\rangle$
$t_C$	$\frac{\partial t_C}{\partial p_\omega^1} = (1 - k_\omega^{11}) \cdot (t_{A_2} - t_{A_1})$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) < 0 \end{cases}$
	$\frac{\partial t_C}{\partial p_\omega^0} = k_\omega^{00j} \cdot (t_{A_2} - t_{A_1})$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) < 0 \end{cases}$
	$\frac{\partial t_C}{\partial k_\omega^{00j}} = p_\omega^0 \cdot (t_{A_2} - t_{A_1})$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (t_{A_2} - t_{A_1}) < 0 \end{cases}$
	$\frac{\partial t_C}{\partial k_\omega^{11}} = p_\omega^1 \cdot (t_{A_1} - t_{A_2})$	$\begin{cases} \langle\langle + \rangle\rangle, \text{якщо } (t_{A_1} - t_{A_2}) \geq 0 \\ \langle\langle - \rangle\rangle, \text{якщо } (t_{A_1} - t_{A_2}) < 0 \end{cases}$

За знаками частинних похідних показників надійності (табл. 6.3) згідно з кроком 5 методики узагальнення отримаємо нечіткі моделі надійності (6.16)-(6.18).

*Твердження 6.3.* При нечітких ймовірнісно-часових характеристиках оператора та з логічної умови алгоритмічна структура « $\omega$ -ітерація»  $D = \{A\}$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_D^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{D_\alpha}^1, \bar{p}_{D_\alpha}^1 \right), \quad (6.23)$$

$$\text{де } \underline{p}_D^1 = \frac{\underline{p}_A^1 \cdot \underline{k}_\omega^{11}}{1 - [\underline{p}_A^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1}^m \underline{p}_A^{0j} \cdot \underline{k}_\omega^{00j}]}; \quad \bar{p}_D^1 = \min \left[ 1, \frac{\bar{p}_A^1 \cdot \bar{k}_\omega^{11}}{1 - [\bar{p}_A^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1}^m \bar{p}_A^{0j} \cdot \bar{k}_\omega^{00j}]} \right];$$

$$\tilde{p}_D^{0j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{D_\alpha}^{0j}, \bar{p}_{D_\alpha}^{0j} \right), \quad j = \overline{1, m}, \quad (6.24)$$

$$\text{де } \underline{p}_D^{0j} = \frac{\underline{p}_A^{0j} \cdot (1 - \bar{k}_\omega^{00j})}{1 - [\underline{p}_A^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1}^m \underline{p}_A^{0j} \cdot \bar{k}_\omega^{00j}]}; \quad \bar{p}_D^{0j} = \frac{\bar{p}_A^{0j} \cdot (1 - \underline{k}_\omega^{00j})}{1 - [\bar{p}_A^1 \cdot (1 - \underline{k}_\omega^{11}) + \sum_{j=1}^m \bar{p}_A^{0j} \cdot \underline{k}_\omega^{00j}]};$$

$$\tilde{t}_D = \bigcup_{\alpha \in [0,1]} \left( \underline{t}_{D_\alpha}, \bar{t}_{D_\alpha} \right), \quad (6.25)$$

$$\underline{t}_D = \frac{\underline{t}_A + \underline{t}_\omega}{1 - [\underline{p}_A^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1}^m \underline{p}_A^{0j} \cdot \underline{k}_\omega^{00j}]}; \quad \bar{t}_D = \frac{\bar{t}_A + \bar{t}_\omega}{1 - [\bar{p}_A^1 \cdot (1 - \underline{k}_\omega^{11}) + \sum_{j=1}^m \bar{p}_A^{0j} \cdot \bar{k}_\omega^{00j}]}.$$

*Доведення.* Початкова математична модель ймовірнісних характеристик структури « $\omega$ -ітерація» є такою [79]

$$\mathbf{P}_D = \mathbf{P}_Z \cdot \mathbf{P}_A \cdot (\mathbf{I} - \mathbf{K}_\omega^0 \cdot \mathbf{P}_Z \cdot \mathbf{P}_A)^{-1} \cdot \mathbf{K}_\omega^1. \quad (6.26)$$

Перепишемо вираз (6.26) таким чином

$$\begin{aligned}
\mathbf{P}_D &= \mathbf{P}_Z \cdot \mathbf{P}_A \cdot \left[ \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} k_\omega^{10} & 0 & \dots & 0 \\ 0 & k_\omega^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{00_m} \end{pmatrix} \cdot \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ \dots & \dots & \dots & \dots \\ p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \end{pmatrix} \right]^{-1} \cdot \mathbf{K}_\omega^1 = \\
&= \mathbf{P}_Z \cdot \mathbf{P}_A \cdot \begin{pmatrix} 1-p_A^1 \cdot k_\omega^{10} & -p_A^{0_1} \cdot k_\omega^{10} & \dots & -p_A^{0_m} \cdot k_\omega^{10} \\ -p_A^1 \cdot k_\omega^{00_1} & 1-p_A^{0_1} \cdot k_\omega^{00_1} & \dots & -p_A^{0_m} \cdot k_\omega^{00_1} \\ \dots & \dots & \dots & \dots \\ -p_A^1 \cdot k_\omega^{00_m} & -p_A^{0_1} \cdot k_\omega^{00_m} & \dots & 1-p_A^{0_m} \cdot k_\omega^{00_m} \end{pmatrix}^{-1} \cdot \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01_m} \end{pmatrix} = \mathbf{P}_Z \cdot \mathbf{P}_A \cdot \\
&\cdot \frac{1}{1-p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0_j} \cdot k_\omega^{00_j}} \cdot \begin{pmatrix} 1 - \sum_{j=1,m} p_A^{0_j} \cdot k_\omega^{00_j} & p_A^{0_1} \cdot k_\omega^{10} & \dots & p_A^{0_m} \cdot k_\omega^{10} \\ p_A^1 \cdot k_\omega^{00_1} & 1-p_A^1 \cdot k_\omega^{10} - p_A^{0_m} \cdot k_\omega^{00_m} & \dots & p_A^{0_m} \cdot k_\omega^{00_1} \\ \dots & \dots & \dots & \dots \\ p_A^1 \cdot k_\omega^{00_m} & p_A^{0_1} \cdot k_\omega^{00_m} & \dots & 1-p_A^1 \cdot k_\omega^{10} - p_A^{0_1} \cdot k_\omega^{00_1} \end{pmatrix} \cdot \\
&\cdot \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01_2} \end{pmatrix} = \frac{1}{1-p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0_j} \cdot k_\omega^{00_j}} \cdot \begin{pmatrix} p_A^1 \cdot k_\omega^{11} & p_A^{0_1} \cdot k_\omega^{01_1} & \dots & p_A^{0_m} \cdot k_\omega^{01_m} \\ p_A^1 \cdot k_\omega^{11} & p_A^{0_1} \cdot k_\omega^{01_1} & \dots & p_A^{0_m} \cdot k_\omega^{01_m} \\ \dots & \dots & \dots & \dots \\ p_A^1 \cdot k_\omega^{11} & p_A^{0_1} \cdot k_\omega^{01_1} & \dots & p_A^{0_m} \cdot k_\omega^{01_m} \end{pmatrix}
\end{aligned}$$

З останньої матриці запишемо ймовірнісні характеристики структури « $\omega$ -ітерація»:

$$p_D^1 = \frac{p_A^1 \cdot k_\omega^{11}}{1 - [p_A^1 \cdot (1 - k_\omega^{11}) + \sum_{j=1,m} p_A^{0_j} \cdot k_\omega^{00_j}]}; \quad (6.27)$$

$$p_D^{0_j} = \frac{p_A^{0_j} \cdot (1 - k_\omega^{00_j})}{1 - [p_A^1 \cdot (1 - k_\omega^{11}) + \sum_{j=1,m} p_A^{0_j} \cdot k_\omega^{00_j}]}, \quad j = \overline{1, m}. \quad (6.28)$$

Початкова модель середнього часу виконання структури « $\omega$ -ітерація» визначається за такою формулою [79]:

$$t_D = \frac{t_\omega + t_A}{1 - b}. \quad (6.29)$$

Для узагальнення моделей (6.27)–(6.29) на випадок нечітких початкових даних в табл. 6.4 представлені частинні похідні та їх знаки.

Таблиця 6.4

Знаки частинних похідних показників надійності структури « $\omega$ -ітерація»

Показник	Частинна похідна	Знак похідної
$p_D^1$	$\frac{\partial p_D^1}{\partial p_A^1} = \frac{k_\omega^{11} \cdot (p_A^1 + \sum_{j=1,m} p_A^{0j} \cdot (1 - k_\omega^{00j}))}{(1 - p_A^1 \cdot (1 - k_\omega^{11}) - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial p_D^1}{\partial p_A^{0j}} = \frac{p_A^1 \cdot k_\omega^{11} \cdot (1 - k_\omega^{00j})}{(1 - p_A^1 \cdot (1 - k_\omega^{11}) - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
$p_D^1$	$\frac{\partial p_D^1}{\partial k_\omega^{11}} = \frac{p_A^1 \cdot \sum_{j=1,m} p_A^{0j} \cdot (1 - k_\omega^{00j})}{(1 - p_A^1 \cdot (1 - k_\omega^{11}) - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial p_D^1}{\partial k_\omega^{00j}} = \frac{p_A^1 \cdot p_A^{0j} \cdot k_\omega^{11}}{(1 - p_A^1 \cdot (1 - k_\omega^{11}) - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
$p_D^{0j},$ $j = \overline{1,m}$	$\frac{\partial p_D^{0j}}{\partial p_A^{0j}} = \frac{k_\omega^{11}}{(1 - p_A^1 \cdot (1 - k_\omega^{11}) - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial p_D^{0j}}{\partial p_A^1} = \frac{k_\omega^{01j} \cdot (p_A^1 \cdot k_\omega^{11} + p_A^{0j} \cdot k_\omega^{00j} + \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{01j})}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial p_D^{0j}}{\partial k_\omega^{11}} = \frac{-p_A^1}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«-»
	$\frac{\partial p_D^{0j}}{\partial k_\omega^{00j}} = \frac{-p_A^{0j} \cdot (p_A^1 \cdot k_\omega^{11} + p_A^{0j} \cdot k_\omega^{01j} + \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{01j})}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«-»

Продовження табл. 6.4

Показник	Частинна похідна	Знак похідної
$t_D$	$\frac{\partial t_D}{\partial p_A^1} = \frac{\partial t_D}{\partial t_\omega} = \frac{(p_A^1 \cdot k_\omega^{11} + \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{01j})}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial t_D}{\partial p_A^1} = \frac{(t_A + t_\omega) \cdot k_\omega^{10}}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
$t_D$	$\frac{\partial t_D}{\partial p_A^{0j}} = \frac{(t_A + t_\omega) \cdot k_\omega^{00j}}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»
	$\frac{\partial t_D}{\partial k_\omega^{11}} = \frac{-(t_A + t_\omega) \cdot p_A^1}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«-»
	$\frac{\partial t_D}{\partial k_\omega^{00j}} = \frac{(t_A + t_\omega) \cdot p_A^{0j}}{(1 - p_A^1 \cdot k_\omega^{10} - \sum_{j=1,m} p_A^{0j} \cdot k_\omega^{00j})^2}$	«+»

За знаками частинних похідних показників надійності (табл. 6.4) згідно з кроком 5 методики узагальнення отримуємо нечіткі моделі надійності (6.23)–(6.25).

*Твердження 6.4.* При нечітких ймовірнісно-часових характеристиках операторів та логічної умови алгоритмічна структура «робота-контроль-добробка»  $F = A(E \vee U)$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_F^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{F_\alpha}^1, \overline{p}_{F_\alpha}^1 \right), \quad (6.30)$$

де  $\underline{p}_F^1 = \underline{p}_A^1 + \sum_{j=1,m} \underline{p}_A^{0j} \cdot \underline{k}_\omega^{00j} \cdot \underline{v}_U^1$ ;  $\overline{p}_F^1 = \min(1, \overline{p}_A^1 + \sum_{j=1,m} \overline{p}_A^{0j} \cdot \overline{k}_\omega^{00j} \cdot \overline{v}_U^1)$ ;

$$\tilde{p}_F^{0j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{F_\alpha}^{0j}, \overline{p}_{F_\alpha}^{0j} \right), \quad j = \overline{1, m}, \quad (6.31)$$

$$\text{де } \underline{p}_F^0 = \underline{p}_A^0 \cdot (1 - \overline{k}_\omega^{00} \cdot \underline{v}_U^1); \quad \overline{p}_F^0 = \overline{p}_A^0 \cdot (1 - \underline{k}_\omega^{00} \cdot \underline{v}_U^1);$$

$$\tilde{t}_F = \bigcup_{\alpha \in [0,1]} (\underline{t}_{F\alpha}, \overline{t}_{F\alpha}), \quad (6.32)$$

$$\text{де } \underline{t}_F = \underline{t}_A + \underline{t}_\omega + \underline{t}_U \cdot (\underline{p}_A^1 \cdot (1 - \overline{k}_\omega^{11}) + \sum_{j=1,m} \underline{p}_A^0 \cdot \underline{k}_\omega^{00j});$$

$$\overline{t}_F = \overline{t}_A + \overline{t}_\omega + \overline{t}_U \cdot (\overline{p}_A^1 \cdot (1 - \underline{k}_\omega^{11}) + \sum_{j=1,m} \overline{p}_A^0 \cdot \overline{k}_\omega^{00j}).$$

*Доведення.* Початкова математична модель ймовірнісних характеристик структури «робота-контроль-доробка» є такою [79]

$$\mathbf{P}_F = \mathbf{P}_A \cdot (\mathbf{K}_\omega^1 + \mathbf{K}_\omega^0 \cdot \mathbf{P}_U). \quad (6.33)$$

Перепишемо вираз (6.33) таким чином:

$$\begin{aligned} \mathbf{P}_F &= \begin{pmatrix} p_A^1 & p_A^0 & \dots & p_A^m \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \left[ \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01m} \end{pmatrix} + \begin{pmatrix} k_\omega^{10} & 0 & \dots & 0 \\ 0 & k_\omega^{00} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{00m} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \dots & 0 \\ v_U^1 & v_U^0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_U^m & 0 & \dots & v_U^0 \end{pmatrix} \right] = \\ &= \begin{pmatrix} p_A^1 & p_A^0 & \dots & p_A^m \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \dots & 0 \\ k_\omega^{00} \cdot v_U^1 & k_\omega^{00} \cdot v_U^0 + k_\omega^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ k_\omega^{00m} \cdot v_U^1 & 0 & \dots & k_\omega^{00m} \cdot v_U^0 + k_\omega^{01m} \end{pmatrix} = \\ &= \begin{pmatrix} p_A^1 + \sum_{j=1}^m p_A^0 \cdot k_\omega^{00j} \cdot v_U^1 & p_A^0 \cdot (k_\omega^{01} + k_\omega^{00} \cdot v_U^0) & \dots & p_A^m \cdot (k_\omega^{01m} + k_\omega^{00m} \cdot v_U^0) \\ k_\omega^{00} \cdot v_U^1 & k_\omega^{01} + k_\omega^{00} \cdot v_U^0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ k_\omega^{00m} \cdot v_U^1 & 0 & \dots & k_\omega^{01m} + k_\omega^{00m} \cdot v_U^0 \end{pmatrix} \end{aligned}$$

З останньої матриці запишемо ймовірнісні характеристики структури «робота-контроль-доробка»:

$$p_F^1 = p_A^1 + \sum_{j=1,m} p_A^0 \cdot k_\omega^{00j} \cdot v_U^1, \quad (6.34)$$

$$p_F^0 = p_A^0 \cdot \left(1 - k_\omega^{00j} \cdot v_U^1\right), \quad j = \overline{1,m}. \quad (6.35)$$

Початкова модель середнього часу виконання структури «робота-контроль-доробка» визначається за такою формулою [79]

$$t_F = t_A + t_\omega + t_U \cdot \left( p_A^1 \cdot k_\omega^{10} + \sum_{j=1}^m p_A^{0j} \cdot k_\omega^{00j} \right). \quad (6.36)$$

Для узагальнення моделей (6.34)–(6.36) на випадок нечітких початкових даних в табл. 6.5 представлені частинні похідні та їх знаки.

Таблиця 6.5

Знаки частинних похідних показників структури  
«робота-контроль-доробка»

Показник	Частинна похідна	Знак похідної
$p_F^1$	$\frac{\partial p_F^1}{\partial p_A^1} = 1; \quad \frac{\partial p_F^1}{\partial p_A^{0j}} = k_\omega^{00j} \cdot v_U^{1j};$ $\frac{\partial p_F^1}{\partial k_\omega^{00j}} = p_A^{0j} \cdot v_U^{1j}; \quad \frac{\partial p_F^1}{\partial v_U^{1j}} = p_A^{0j} \cdot k_\omega^{00j}$	«+»
$p_{F^j}^0, \overline{j=1, m}$	$\frac{\partial p_{F^j}^0}{\partial p_A^{0j}} = 1 - k_\omega^{00j} \cdot v_U^{1j}$ $\frac{\partial p_{F^j}^0}{\partial k_\omega^{00j}} = -p_A^{0j} \cdot v_U^{1j}; \quad \frac{\partial p_{F^j}^0}{\partial v_U^{1j}} = -p_A^{0j} \cdot k_\omega^{00j}$	«+» «-»
$t_F$	$\frac{\partial t_F}{\partial t_A} = 1; \quad \frac{\partial t_F}{\partial t_\omega} = 1; \quad \frac{\partial t_F}{\partial t_U} = p_A^1 \cdot k_\omega^{10} + \sum_{j=1, m} p_A^{0j} \cdot k_\omega^{00j}$	«+»
	$\frac{\partial t_F}{\partial p_A^1} = t_U \cdot k_\omega^{10}; \quad \frac{\partial t_F}{\partial p_A^{0j}} = t_U \cdot k_\omega^{00j};$ $\frac{\partial t_F}{\partial k_\omega^{00j}} = t_U \cdot p_A^{0j} \quad \frac{\partial t_F}{\partial k_\omega^{11}} = -t_U \cdot p_A^1$	«+» «-»

За знаками частинних похідних показників надійності (табл. 6.5) згідно з кроком 5 методики узагальнення отримуємо нечіткі моделі надійності (6.30)–(6.32).



*Твердження 6.5.* При нечітких ймовірнісно-часових характеристиках оператора алгоритмічна структура «багаторазова робота»  $G = A^M$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_G^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{G\alpha}^1, \bar{p}_{G\alpha}^1 \right), \quad (6.37)$$

де  $\underline{p}_G^1 = (\underline{p}_A^1)^M$ ;  $\bar{p}_G^1 = (\bar{p}_A^1)^M$ ;

$$\tilde{p}_G^{0_j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{G\alpha}^{0_j}, \bar{p}_{G\alpha}^{0_j} \right), \quad j = \overline{1, m}, \quad (6.38)$$

де  $\underline{p}_G^{0_j} = \frac{\underline{p}_A^{0_j} \cdot (1 - (\underline{p}_A^1)^M)}{1 - \underline{p}_A^1}$ ;  $\bar{p}_G^{0_j} = \frac{\bar{p}_A^{0_j} \cdot (1 - (\bar{p}_A^1)^M)}{1 - \bar{p}_A^1}$ ;

$$\tilde{t}_G = \bigcup_{\alpha \in [0,1]} \left( \underline{t}_{G\alpha}, \bar{t}_{G\alpha} \right), \quad (6.39)$$

де  $\underline{t}_G = M \cdot \underline{t}_A$ ;  $\bar{t}_G = M \cdot \bar{t}_A$ .

*Доведення.* Початкова математична модель ймовірнісних характеристик структури «багаторазова робота» є такою [79]

$$\mathbf{P}_G = (\mathbf{P}_A)^M. \quad (6.40)$$

Перепишемо вираз (6.40) таким чином

$$\begin{aligned} \mathbf{P}_G &= \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}^M = \begin{pmatrix} (p_A^1) & p_A^{0_1} \cdot (1 + p_A^1) & \dots & p_A^{0_m} \cdot (1 + p_A^1) \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}^{M-2} = \\ &= \begin{pmatrix} (p_A^1)^3 & p_A^{0_1} \cdot (1 + p_A^1 + (p_A^1)^2) & \dots & p_A^{0_m} \cdot (1 + p_A^1 + (p_A^1)^2) \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}^{M-3} = \dots = \end{aligned}$$

$$= \begin{pmatrix} (p_A^1)^M & p_A^{0_1} \cdot \frac{1 - (p_A^1)^M}{1 - p_A^1} & \dots & p_A^{0_m} \cdot \frac{1 - (p_A^1)^M}{1 - p_A^1} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

З останньої матриці запишемо ймовірнісні характеристики структури «багаторазова робота»:

$$p_G^1 = (p_A^1)^M, \quad (6.41)$$

$$p_G^{0_j} = \frac{p_A^{0_j} \cdot (1 - (p_A^1)^M)}{1 - p_A^1}, \quad j = \overline{1, m}. \quad (6.42)$$

Початкова модель середнього часу виконання структури «багаторазова робота» визначається за такою формулою [79]

$$t_G = M \cdot t_A. \quad (6.43)$$

Для узагальнення моделей (6.41)–(6.43) на випадок нечітких початкових даних в табл. 6.6 представлені частинні похідні та їх знаки.

Таблиця 6.6

Знаки частинних похідних показників надійності структури  
«багаторазова робота»

Показник	Частинна похідна	Знак похідної
$p_G^1$	$\frac{\partial p_G^1}{\partial p_A^1} = M \cdot (p_A^1)^{M-1};$	«+»
$p_G^{0_j},$ $j = \overline{1, m}$	$\frac{\partial p_G^{0_j}}{\partial p_A^{0_j}} = \frac{1 - (p_A^1)^M}{1 - p_A^1}$	«+»
	$\frac{\partial p_G^{0_j}}{\partial p_A^1} = \frac{p_A^{0_j}}{1 - p_A^1} \cdot \left( \frac{1 - (p_A^1)^M}{1 - p_A^1} - M \cdot (p_A^1)^{M-1} \right)$	«+»
$t_G$	$\frac{\partial t_G}{\partial t_A} = M$	«+»

За знаками частинних похідних показників надійності (табл. 6.6) згідно з кроком 5 методики узагальнення отримаємо нечіткі моделі надійності (6.37)-(6.39).

*Твердження 6.6.* При нечітких ймовірнісно-часових характеристиках оператора та логічної умови алгоритмічна структура « $\omega$ -доробка»  $L = \{U\}$  може бути замінена єдиним робочим оператором з такими еквівалентними характеристиками надійності:

$$\tilde{p}_L^1 = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{L\alpha}^1, \bar{p}_{L\alpha}^1 \right), \quad (6.44)$$

$$\text{де } \underline{p}_L^1 = \underline{p}_\omega^1 + \sum_{j=1,m} \frac{\underline{p}_\omega^{0j} \cdot \underline{v}_U^1 \cdot \underline{k}_\omega^{00j}}{1 - (1 - \underline{v}_U^1) \cdot \underline{k}_\omega^{00j}}; \quad \bar{p}_L^1 = \bar{p}_\omega^1 + \sum_{j=1,m} \frac{\bar{p}_\omega^{0j} \cdot \bar{v}_U^1 \cdot \bar{k}_\omega^{00j}}{1 - (1 - \bar{v}_U^1) \cdot \bar{k}_\omega^{00j}};$$

$$\tilde{p}_L^{0j} = \bigcup_{\alpha \in [0,1]} \left( \underline{p}_{L\alpha}^{0j}, \bar{p}_{L\alpha}^{0j} \right), \quad (6.45)$$

$$\text{де } \underline{p}_L^{0j} = \frac{\underline{p}_\omega^{0j} \cdot (1 - \bar{k}_\omega^{00j})}{1 - (1 - \bar{v}_U^1) \cdot \bar{k}_\omega^{00j}}; \quad \bar{p}_L^{0j} = \frac{\bar{p}_\omega^{0j} \cdot (1 - \underline{k}_\omega^{00j})}{1 - (1 - \underline{v}_U^1) \cdot \underline{k}_\omega^{00j}};$$

$$\tilde{t}_L = \bigcup_{\alpha \in [0,1]} \left( \underline{t}_{L\alpha}, \bar{t}_{L\alpha} \right), \quad (6.46)$$

$$\text{де } \underline{t}_L = \underline{t}_\omega + \frac{\left( \underline{p}_\omega^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1,m} \underline{p}_\omega^{0j} \cdot \underline{k}_\omega^{00j} \right) \cdot (\underline{t}_A + \underline{t}_\omega)}{1 - \left( \underline{p}_\omega^1 \cdot (1 - \bar{k}_\omega^{11}) + \sum_{j=1,m} \underline{p}_\omega^{0j} \cdot \underline{k}_\omega^{00j} \right)};$$

$$\bar{t}_L = \bar{t}_\omega + \frac{\left( \bar{p}_\omega^1 \cdot (1 - \underline{k}_\omega^{11}) + \sum_{j=1,m} \bar{p}_\omega^{0j} \cdot \bar{k}_\omega^{00j} \right) \cdot (\bar{t}_A + \bar{t}_\omega)}{1 - \left( \bar{p}_\omega^1 \cdot (1 - \underline{k}_\omega^{11}) + \sum_{j=1,m} \bar{p}_\omega^{0j} \cdot \bar{k}_\omega^{00j} \right)}.$$

*Доведення.* Початкова математична модель ймовірнісних характеристик структури « $\omega$ -доробка» задається такою формулою [79]:

$$\mathbf{P}_L = \mathbf{P}_\omega \cdot (\mathbf{I} - \mathbf{K}_\omega^0 \cdot \mathbf{P}_U)^{-1} \cdot \mathbf{K}_\omega^1. \quad (6.47)$$

Перепишемо вираз (6.47) таким чином

$$\begin{aligned}
 \mathbf{P}_L &= \begin{pmatrix} p_\omega^1 & p_\omega^0 & \dots & p_\omega^m \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{I} - \begin{pmatrix} k_\omega^{10} & 0 & \dots & 0 \\ 0 & k_\omega^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{00_m} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \dots & 0 \\ v_U^1 & v_U^0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_U^m & 0 & \dots & v_U^0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01_m} \end{pmatrix} = \\
 &= \begin{pmatrix} p_\omega^1 & p_\omega^0 & \dots & p_\omega^m \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ -v_U^1 \cdot k_\omega^{00_1} & 1 - v_U^1 \cdot k_\omega^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ -v_U^m \cdot k_\omega^{00_m} & 0 & \dots & 1 - v_U^m \cdot k_\omega^{00_m} \end{pmatrix}^{-1} \cdot \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01_m} \end{pmatrix} = \\
 &= \begin{pmatrix} p_\omega^1 + \sum_{j=1, m} \frac{p_\omega^0 \cdot v_U^1 \cdot k_\omega^{00_j}}{1 - v_U^0 \cdot k_\omega^{00_j}} & \frac{p_\omega^0 \cdot k_\omega^{01_1}}{1 - v_U^0 \cdot k_\omega^{00_1}} & \dots & \frac{p_\omega^0 \cdot k_\omega^{01_m}}{1 - v_U^0 \cdot k_\omega^{00_m}} \\ \frac{v_U^1 \cdot k_\omega^{00_1}}{1 - v_U^0 \cdot k_\omega^{00_1}} & \frac{k_\omega^{01_1}}{1 - v_U^0 \cdot k_\omega^{00_1}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \frac{v_U^m \cdot k_\omega^{00_m}}{1 - v_U^0 \cdot k_\omega^{00_m}} & 0 & \dots & \frac{k_\omega^{01_m}}{1 - v_U^0 \cdot k_\omega^{00_m}} \end{pmatrix} \cdot
 \end{aligned}$$

З останньої матриці запишемо ймовірнісні характеристики структури « $\omega$ -доробка»:

$$p_L^1 = p_\omega^1 + \sum_{j=1}^m \frac{p_\omega^0 \cdot v_U^1 \cdot k_\omega^{00_j}}{1 - \left(1 - v_U^1\right) \cdot k_\omega^{00_j}}; \quad (6.48)$$

$$p_L^{0_j} = \frac{p_\omega^0 \cdot (1 - k_\omega^{00_j})}{1 - \left(1 - v_U^1\right) \cdot k_\omega^{00_j}}, \quad j = \overline{1, m}. \quad (6.49)$$

Початкова модель середнього часу виконання структури « $\omega$ -доробка» визначається за такою формулою [79]

$$t_L = t_\omega + b \cdot \frac{t_\omega + t_U}{1 - b}. \quad (6.50)$$

Для узагальнення моделей (6.48)–(6.50) для нечітких початкових даних в табл. 6.7 представлені частинні похідні та їх знаки.

Таблиця 6.7

Знаки частинних похідних показників надійності  
структури « $\omega$ -доробка»

Параметр	Частинна похідна	Знак похідної
$p_L^1$	$\frac{\partial p_L^1}{\partial p_\omega^0} = \frac{v_U^1 \cdot k_\omega^{00j}}{1 - v_U^0 \cdot k_\omega^{00j}}; \quad \frac{\partial p_L^1}{\partial v_U^1} = \frac{p_\omega^0 \cdot k_\omega^{00j} \cdot k_\omega^{01j}}{(1 - v_U^0 \cdot k_\omega^{00j})^2};$ $\frac{\partial p_L^1}{\partial p_\omega^1} = 1; \quad \frac{\partial p_L^1}{\partial k_\omega^{00j}} = \frac{p_\omega^0 \cdot v_U^1}{(1 - v_U^0 \cdot k_\omega^{00j})^2}$	«+»
$p_L^0, \overline{j=1,m}$	$\frac{\partial p_L^0}{\partial p_\omega^0} = \frac{k_\omega^{01j}}{1 - v_U^0 \cdot k_\omega^{00j}};$ $\frac{\partial p_L^0}{\partial k_\omega^{00j}} = -\frac{p_\omega^0 \cdot v_U^1}{(1 - v_U^0 \cdot k_\omega^{00j})^2}; \quad \frac{\partial p_L^0}{\partial v_U^1} = -\frac{p_\omega^0 \cdot k_\omega^{00j} \cdot k_\omega^{01j}}{(1 - v_U^0 \cdot k_\omega^{00j})^2}$	«+» «-»
$t_L$	$\frac{\partial t_L}{\partial p_\omega^1} = \frac{k_\omega^{10} \cdot (t_\omega + t_U)}{(p_\omega^1 \cdot k_\omega^{11} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{01j})^2}$	«+»
	$\frac{\partial t_L}{\partial p_\omega^0} = \frac{k_\omega^{00j} \cdot (t_\omega + t_U)}{(p_\omega^1 \cdot k_\omega^{11} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{01j})^2}$	«+»
	$\frac{\partial t_L}{\partial k_\omega^{11}} = -\frac{p_\omega^1 \cdot (t_\omega + t_U)}{(p_\omega^1 \cdot k_\omega^{11} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{01j})^2}$	«-»
	$\frac{\partial t_L}{\partial k_\omega^{00j}} = \frac{p_\omega^0 \cdot (t_\omega + t_U)}{(p_\omega^1 \cdot k_\omega^{11} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{01j})^2}$	«+»
	$\frac{\partial t_L}{\partial t_\omega} = 1 + \frac{\partial t_L}{\partial t_U} = 1 + \frac{p_\omega^1 \cdot k_\omega^{10} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{00j}}{p_\omega^1 \cdot k_\omega^{00j} + \sum_{j=1,m} p_\omega^0 \cdot k_\omega^{01j}}$	«+»

За знаками частинних похідних показників надійності (табл. 6.7) згідно з кроком 5 методики узагальнення отримаємо нечіткі моделі надійності (6.44)–(6.46).

### 6.5. Алгоритм аналізу надійності багатовимірних алгоритмічних процесів при нечітких початкових даних

Нечіткі характеристики надійності багатовимірних АП пропонується розраховувати за алгоритмом, який наведено нижче. Він узагальнює алгоритм нечіткого аналізу надійності АП [82] на випадок багатовимірної концепції врахування помилок.

*⟨Задати структуру АП у вигляді суперпозиції  $M$  алгоритмічних структур ⟩;*  
*⟨Задати за формулою (6.1) початкові ймовірнісно-часові характеристики операторів та логічних умов ⟩;*  
*⟨Перетворити початкові характеристики в  $\alpha$ -форми за формулами з табл. 6.1 ⟩;*  
*⟨Встановити лічильник алгоритмічних структур:  $i:=1$  ⟩;*  
**While** *⟨Виконується умова  $i < M$  ⟩* **do**  
**If** *⟨ $i$ -та структура «послідовна»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.9)–(6.11)⟩;*  
**End;**  
**If** *⟨ $i$ -та структура « $\omega$ -диз'юнкція»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.16)–(6.18)⟩;*  
**End;**  
**If** *⟨ $i$ -та структура « $\omega$ -ітерація»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.23)–(6.25)⟩;*  
**End;**  
**If** *⟨ $i$ -та структура «робота–контроль–добробка»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.30)–(6.32)⟩;*  
**End;**  
**If** *⟨ $i$ -та структура «багаторазова робота»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.37)–(6.39)⟩;*  
**End;**  
**If** *⟨ $i$ -та структура « $\omega$ -добробка»? ⟩*  
*⟨Замінити  $i$ -ту структуру єдиним оператором за формулами (6.44)–(6.46)⟩;*  
**End;**  
**End.**

Наведений алгоритм перетворює довільний багатовимірний АП до єдиного робочого оператора з еквівалентними нечіткими характеристиками часу виконання, ймовірності безпомилкового виконання та ймовірностей наявності помилок різних типів.

Параметри функцій належностей, якими формалізуються ймовірнісно-часові характеристики операторів та логічних умов, впливають на значення показників надійності АП по-різному. Одні параметри мають значний вплив, а інші незначний. Для визначення впливу параметрів на практиці, як правило, застосовують аналіз чутливості, який дозволяє визначити діапазони чутливості показників надійності до варіації цих параметрів [69]. Якщо діапазон чутливості є суттєвим, то при проектуванні АП необхідно звернути увагу на точність експериментального визначення впливових параметрів. Нижче наводиться ітераційний алгоритм аналізу чутливості:

```

<Позначити параметр функції належності через  $a$ >;
<Задати для параметра діапазон його зміни  $[\underline{a}, \bar{a}]$ >;
<Задати крок варіювання  $h$ >;
<Присвоїти  $a := \underline{a}$ >;
While < Виконується умова  $a < \bar{a}$  > do
    < Обчислити показник надійності АП >;
    < Присвоїти  $a := a + h$  >;
End;
< Визначити діапазон чутливості показника надійності >.

```

**Приклад 6.1.** Розглянемо такий АП:

$$Y = A_1 ( E \vee U ) ( A_2 \vee A_3 ) \{ A_4 \} A_5^5. \quad (6.51)$$

$\omega_1 \qquad \omega_2 \qquad \omega_3$

Нечіткі характеристики надійності операторів та логічних умов алгоритму (6.51) наведені в табл. 6.8–6.10. При виконанні алгоритму (6.51) можливі 5 станів: «0» – відсутність помилок; «1» – помилка 1-го типу; «2» – помилка 2-го типу; «3» – помилка 3-го типу; «4» – помилка 4-го типу. Для розрахунку показників надійності АП (6.53) виконаємо такі кроки:

*Крок 1.* За нечіткими моделями (6.30)–(6.32) розрахуємо еквівалентні характеристики структури  $B_1 = A_1 ( E \vee U )$ :

$$\tilde{p}_{B_1}^1 = (0,9963, 1)_0 \cup (0,9978, 1)_{0,5} \cup (0,9993, 0,9993)_1;$$

$$\begin{aligned} \tilde{p}_{B_1}^{0_1} &= (0, 0,0008)_0 \cup (0,0002, 0,0006)_{0,5} \cup (0,0004, 0,0004)_1; \\ \tilde{p}_{B_1}^{0_2} &= (0,0005, 0,0013)_0 \cup (0,0008, 0,0012)_{0,5} \cup (0,0011, 0,0011)_1; \\ \tilde{p}_{B_1}^{0_3} &= (0,0008, 0,0016)_0 \cup (0,0010, 0,0014)_{0,5} \cup (0,0012, 0,0012)_1; \\ \tilde{p}_{B_1}^{0_4} &= (0,0010, 0,0018)_0 \cup (0,0013, 0,0017)_{0,5} \cup (0,0016, 0,0016)_1; \\ \tilde{t}_{B_1} &= (9,76, 16,33)_0 \cup (10,87, 14,14)_{0,5} \cup (11,99, 11,99)_1; \end{aligned}$$

*Крок 2.* За нечіткими моделями (6.16)–(6.18) розрахуємо еквівалентні характеристики структури  $B_2 = (A_2 \vee A_3)$ :

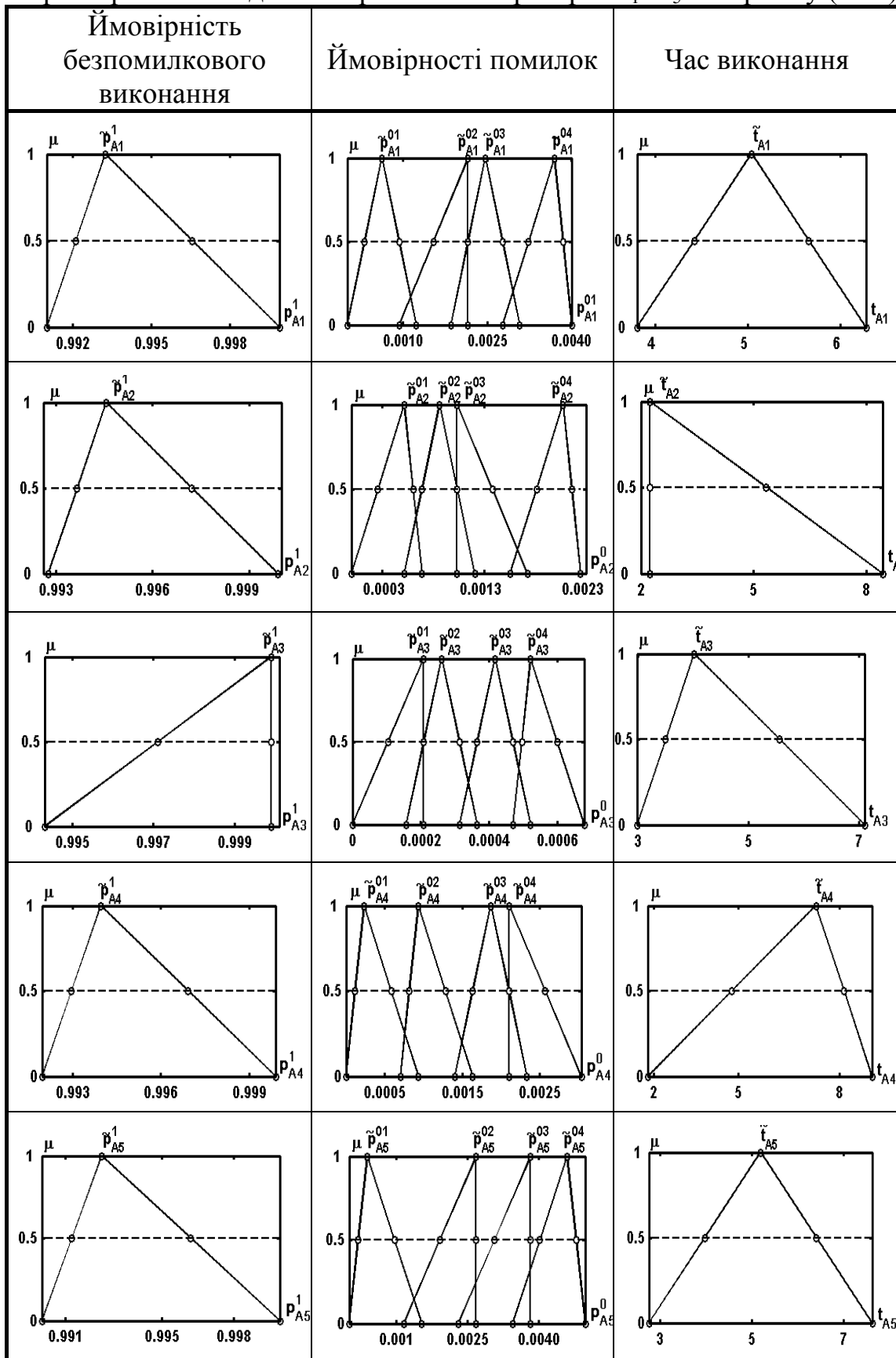
$$\begin{aligned} \tilde{p}_{B_2}^1 &= (0,9262, 1)_0 \cup (0,9521, 0,9892)_{0,5} \cup (0,9781, 0,9781)_1; \\ \tilde{p}_{B_2}^{0_1} &= (0, 0,0132)_0 \cup (0,0018, 0,0084)_{0,5} \cup (0,0037, 0,0037)_1; \\ \tilde{p}_{B_2}^{0_2} &= (0,0006, 0,0137)_0 \cup (0,0023, 0,0088)_{0,5} \cup (0,0040, 0,0040)_1; \\ \tilde{p}_{B_2}^{0_3} &= (0,0011, 0,0142)_0 \cup (0,0027, 0,0092)_{0,5} \cup (0,0042, 0,0042)_1; \\ \tilde{p}_{B_2}^{0_4} &= (0,0016, 0,0146)_0 \cup (0,0034, 0,0099)_{0,5} \cup (0,0052, 0,0052)_1; \\ \tilde{t}_{B_2} &= (8,11, 17,7597)_0 \cup (9,45, 14,31)_{0,5} \cup (10,79, 10,79)_1; \end{aligned}$$

*Крок 3.* За нечіткими моделями (6.23)–(6.25) розрахуємо еквівалентні характеристики структури  $B_3 = \{A_4\}$ :

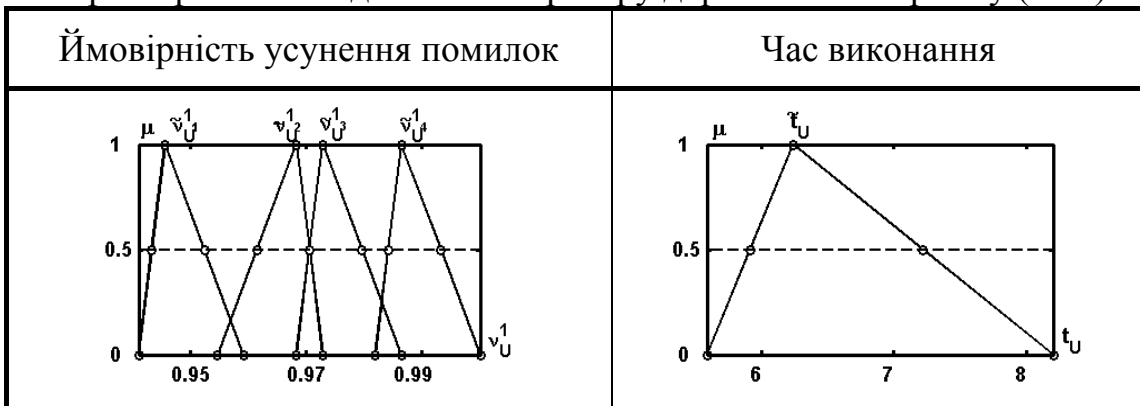
$$\begin{aligned} \tilde{p}_{B_3}^1 &= (0,9941, 1)_0 \cup (0,9955, 1)_{0,5} \cup (0,9969, 0,9969)_1; \\ \tilde{p}_{B_3}^{0_1} &= (0, 0,0006)_0 \cup (0,0001, 0,0003)_{0,5} \cup (0,0001, 0,0001)_1; \\ \tilde{p}_{B_3}^{0_2} &= (0,0003, 0,0009)_0 \cup (0,0004, 0,0007)_{0,5} \cup (0,0005, 0,0005)_1; \\ \tilde{p}_{B_3}^{0_3} &= (0,0004, 0,0010)_0 \cup (0,0006, 0,0009)_{0,5} \cup (0,0008, 0,0008)_1; \\ \tilde{p}_{B_3}^{0_4} &= (0,0004, 0,0010)_0 \cup (0,0006, 0,0009)_{0,5} \cup (0,0007, 0,0007)_1; \\ \tilde{t}_{B_3} &= (8,15, 19,55)_0 \cup (11,36, 17,12)_{0,5} \cup (14,75, 14,75)_1; \end{aligned}$$



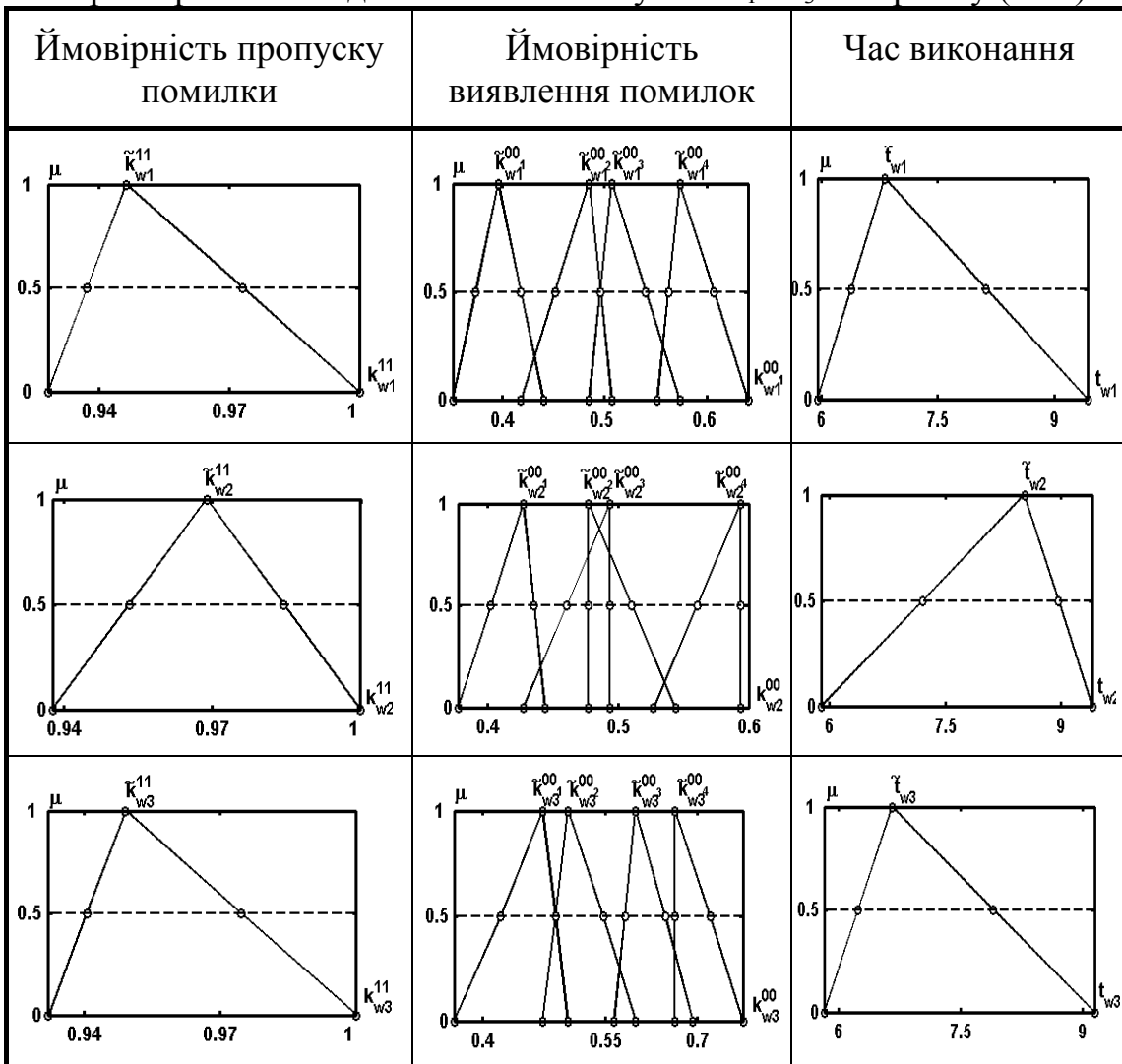
Таблиця 6.8  
Характеристики надійності робочих операторів  $A_1-A_5$  алгоритму (6.51)



Таблиця 6.9  
Характеристики надійності оператора доробки  $U$  алгоритму (6.51)



Таблиця 6.10  
Характеристики надійності логічних умов  $\omega_1-\omega_3$  алгоритму (6.51)



*Крок 4.* За нечіткими моделями (6.37)–(6.39) розрахуємо еквівалентні характеристики структури  $B_4 = A_5^5$ :

$$\tilde{p}_{B_4}^1 = (0,9511, 0,9995)_0 \cup (0,9570, 0,9811)_{0,5} \cup (0,9630, 0,9630)_1;$$

$$\tilde{p}_{B_4}^{0_1} = (0, 0,0077)_0 \cup (0,0009, 0,0048)_{0,5} \cup (0,0019, 0,0019)_1;$$

$$\tilde{p}_{B_4}^{0_2} = (0,0056, 0,0134)_0 \cup (0,0094, 0,0133)_{0,5} \cup (0,0132, 0,0132)_1;$$

$$\tilde{p}_{B_4}^{0_3} = (0,0113, 0,0191)_0 \cup (0,0151, 0,0190)_{0,5} \cup (0,0189, 0,0189)_1;$$

$$\tilde{p}_{B_4}^{0_4} = (0,0169, 0,0249)_0 \cup (0,0198, 0,0238)_{0,5} \cup (0,0226, 0,0226)_1;$$

$$\tilde{t}_{B_4} = (13,79, 38,06)_0 \cup (19,86, 31,99)_{0,5} \cup (25,93, 25,93)_1;$$

*Крок 5.* За нечіткими моделями (6.9)–(6.11) розрахуємо еквівалентні характеристики структури  $B_5 = B_1 \cdot B_2$ :

$$\tilde{p}_{B_5}^1 = (0,9228, 1)_0 \cup (0,9499, 0,9892)_{0,5} \cup (0,9774, 0,9774)_1;$$

$$\tilde{p}_{B_5}^{0_1} = (0, 0,0140)_0 \cup (0,0020, 0,0090)_{0,5} \cup (0,0040, 0,0040)_1;$$

$$\tilde{p}_{B_5}^{0_2} = (0,0010, 0,0150)_0 \cup (0,0031, 0,0101)_{0,5} \cup (0,0052, 0,0052)_1;$$

$$\tilde{p}_{B_5}^{0_3} = (0,0019, 0,0158)_0 \cup (0,0037, 0,0106)_{0,5} \cup (0,0055, 0,0055)_1;$$

$$\tilde{p}_{B_5}^{0_4} = (0,0026, 0,0165)_0 \cup (0,0047, 0,0116)_{0,5} \cup (0,0068, 0,0068)_1;$$

$$\tilde{t}_{B_5} = (17,89, 34,90)_0 \cup (20,33, 28,45)_{0,5} \cup (22,79, 22,79)_1;$$

*Крок 6.* За нечіткими моделями (6.9)–(6.11) розрахуємо еквівалентні характеристики структури  $B_6 = B_5 \cdot B_3$ :

$$\tilde{p}_{B_6}^1 = (0,9173, 1)_0 \cup (0,9456, 1)_{0,5} \cup (0,9743, 0,9743)_1;$$

$$\tilde{p}_{B_6}^{0_1} = (0, 0,0146)_0 \cup (0,0021, 0,0093)_{0,5} \cup (0,0042, 0,0042)_1;$$

$$\tilde{p}_{B_6}^{0_2} = (0,0013, 0,0158)_0 \cup (0,0034, 0,0107)_{0,5} \cup (0,0056, 0,0056)_1;$$

$$\tilde{p}_{B_6}^{0_3} = (0,0023, 0,0168)_0 \cup (0,0042, 0,0115)_{0,5} \cup (0,0062, 0,0062)_1;$$

$$\tilde{p}_{B_6}^0 = (0,0030, 0,0175)_0 \cup (0,0052, 0,0125)_{0,5} \cup (0,0075, 0,0075)_1;$$

$$\tilde{t}_{B_6} = (26,04, 53,64)_0 \cup (31,69, 45,57)_{0,5} \cup (37,53, 37,53)_1;$$

Крок 7. За нечіткими моделями (6.9)–(6.11) розрахуємо еквівалентні характеристики структури  $B_7 = B_6 \cdot B_4$ , які є еквівалентними характеристиками надійності алгоритмічного процесу (6.51):

$$\tilde{p}_Y^1 = (0,8724, 0,9995)_0 \cup (0,9050, 0,9811)_{0,5} \cup (0,9383, 0,9383)_1;$$

$$\tilde{p}_Y^0 = (0, 0,0222)_0 \cup (0,0030, 0,0141)_{0,5} \cup (0,0060, 0,0060)_1;$$

$$\tilde{p}_Y^0 = (0,0064, 0,0292)_0 \cup (0,0123, 0,0240)_{0,5} \cup (0,0185, 0,0185)_1;$$

$$\tilde{p}_Y^0 = (0,0126, 0,0360)_0 \cup (0,0184, 0,0305)_{0,5} \cup (0,0246, 0,0246)_1;$$

$$\tilde{p}_Y^0 = (0,0185, 0,0424)_0 \cup (0,0239, 0,0363)_{0,5} \cup (0,0296, 0,0296)_1;$$

$$\tilde{t}_Y = (39,84, 91,71)_0 \cup (51,56, 77,57)_{0,5} \cup (63,46, 63,46)_1;$$

Отримані оцінки зображені на рис. 6.2. При терм-множині {Низький, Нижче Середнього, Середній, Вище Середнього, Високий} вони інтерпретуються так:

$$\tilde{t}_Y = \langle 39,84, 91,71, \text{Середня} \rangle;$$

$$\tilde{p}_Y^1 = \langle 0,8724, 0,9995, \text{Середня} \rangle;$$

$$\tilde{p}_Y^0 = \langle 0, 0,0222, \text{Низька} \rangle;$$

$$\tilde{p}_Y^0 = \langle 0,0064, 0,0292, \text{Середня} \rangle;$$

$$\tilde{p}_Y^0 = \langle 0,0126, 0,0360, \text{Середня} \rangle;$$

$$\tilde{p}_Y^0 = \langle 0,0185, 0,0424, \text{Середня} \rangle.$$

На рис. 6.3 зображені графіки чутливості показників надійності алгоритму (6.51) до зміни лінгвістичних значень нечітких характеристик операторів  $A_1$  та  $A_5$ . Лінгвістичним значенням відповідають координати максимумів  $\hat{p}_{A_5}^1$ ,  $\hat{p}_{A_1}^1$ ,  $\hat{t}_{A_5}$  та  $\hat{t}_{A_1}$  трикутних функцій належностей нечітких чисел  $\tilde{p}_{A_5}^1$ ,  $\tilde{p}_{A_1}^1$ ,  $\tilde{t}_{A_5}$  та  $\tilde{t}_{A_1}$ .

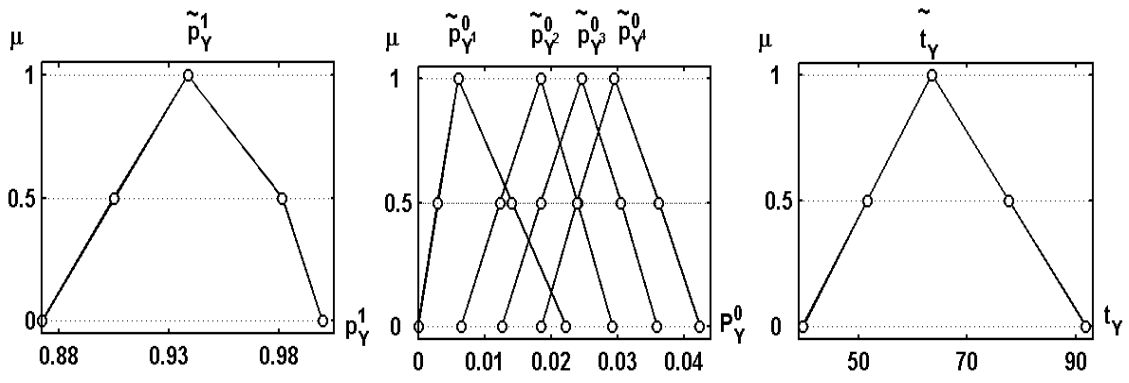


Рис. 6.2. Нечіткі характеристики надійності виконання алгоритму (6.51)

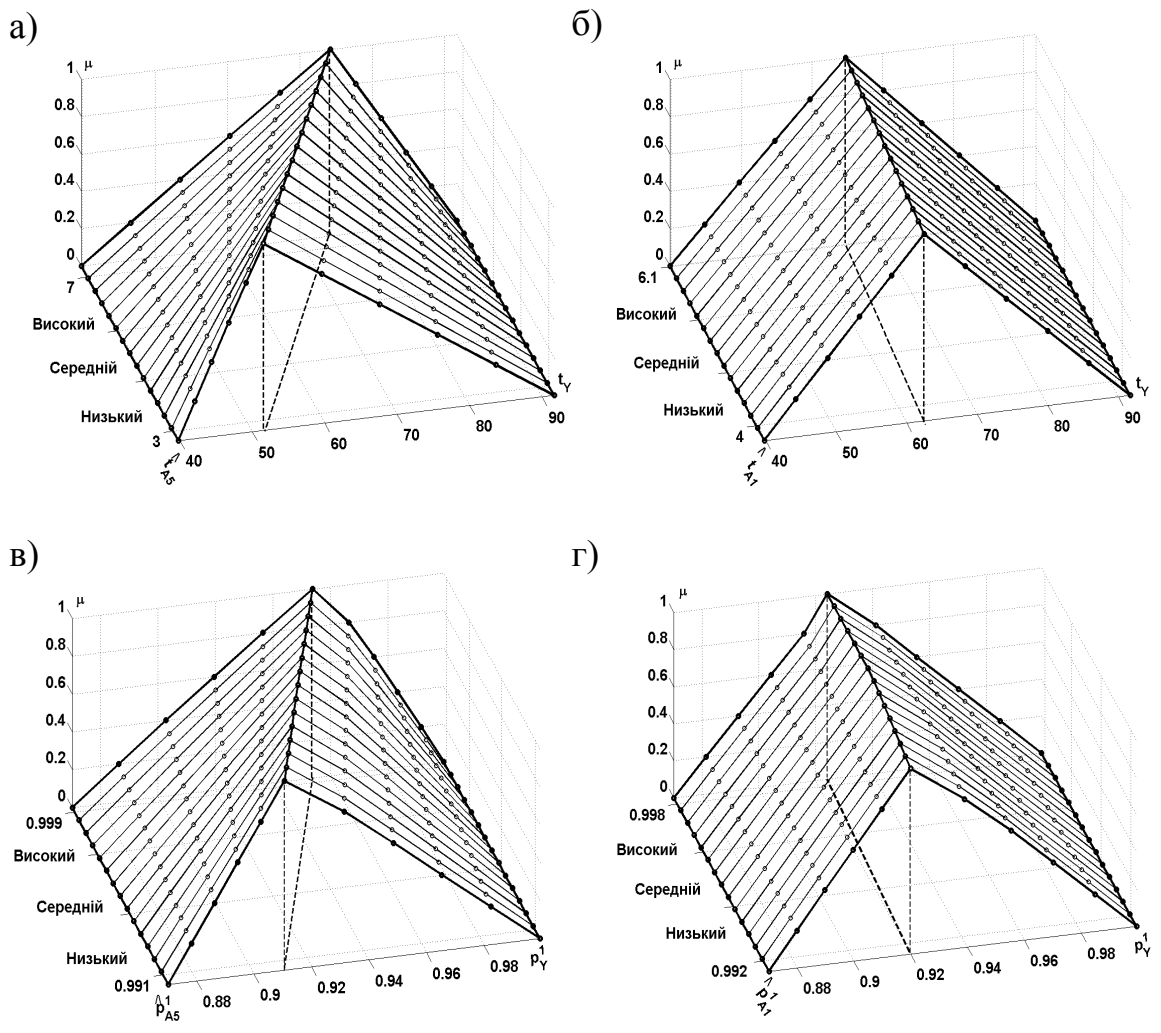


Рис. 6.3. Графіки чутливості: а) нечіткого часу  $\tilde{t}_Y$  до параметра  $\hat{t}_{A5}$ ; б) нечіткого часу  $\tilde{t}_Y$  до параметру  $\hat{t}_{A1}$ ; в) нечіткої ймовірності  $\tilde{p}_Y^1$  до параметру  $\hat{p}_{A5}^1$ ; г) нечіткої ймовірності  $\tilde{p}_Y^1$  до параметру  $\hat{p}_{A1}^1$

З графіків видно, що час ( $\tilde{t}_Y$ ) та ймовірність ( $\tilde{p}_Y^1$ ) виконання алгоритму майже не залежать від лінгвістичних оцінок показників  $\tilde{t}_{A1}$  та  $\tilde{p}_{A1}^1$ , але є чутливими до  $\tilde{t}_{A5}$  та  $\tilde{p}_{A5}^1$ , відповідно. ☒

### 6.6. Моделювання надійності системи «клієнт-банк»

В цьому розділі ілюструється застосування запропонованих нечітких моделей при моделюванні надійності функціонування системи «клієнт-банк», яка є комерційним продуктом компанії «Банк'с Софт & Системс» (<http://www.bssys.com>). Система впроваджена в такі крупні російські банки як «ГАЗПРОМБАНК», «НЕФТЕПРОМБАНК», «ПРОМСТРОЙБАНК», «ФОНДСЕРВИСБАНК», «СОБИНБАНК» та інші.

Система «клієнт-банк» дозволяє організувати обмін документами між клієнтом та банком за безпаперовою технологією. Вона реалізує електронну доставку в банк платіжних доручень, виписок, довідок від клієнта. Однією з задач цієї системи є обробка запиту клієнта по проведенню платіжного доручення через Інтернет. Макет цього запиту показаний на рис. 6.4. Він представляє собою форму, в якій клієнт заповнює відповідні реквізити. Опис реквізитів макета наведено в додатку А.

The screenshot shows a web-based form for creating a payment order. The title is 'Платежное поручение'. The form is organized into several sections:

- Header:** Number (1), Date (2), Payment type (3), and Status (4).
- Payer Information:** Fields for INN (5), KPP (6), and account number (7).
- Bank of Payer:** Fields for bank name (14, 15) and account number (16).
- Payee Information:** Fields for INN (23), KPP (24), and account number (27).
- Bank of Payee:** Fields for bank name (20, 21) and account number (22).
- General Information:** Fields for operation type (30), payment priority (31), and amount (32).
- Document Details:** Fields for KBK (34), OKATO (35), basic payment (36), payment period (37, 38, 39), document number (40), and document date (41, 42, 43).
- Payment Purpose:** Field for the purpose of the payment (45).
- Additional Fields:** Currency code (46) and passport number (47).

At the bottom, there are buttons for 'Сброс' (Reset), 'SearchBIC', and 'Отмена' (Cancel).

Рис. 6.4. Макет платіжного доручення клієнта [3]

Відповідно до [3] обробку клієнтського запиту представимо АП такого виду

$$Y = A_1^{N_1} A_2^{N_2} A_3^{N_3} \left\{ U_1 \right\}_{\omega_1} A_4 A_5 \left( E \vee \{B\} \right)_{\omega_2} A_6 \left( E \vee \{B ( E \vee \{B\} ) \} \right)_{\omega_2 \omega_3}, \quad (6.52)$$

де  $A_1$  – ручне заповнення реквізиту форми документа;

$A_2$  – заповнення реквізиту форми документа з довідника банку;

$A_3$  – заповнення реквізиту форми документа з довідника клієнта;

$N_1, N_2, N_3$  – кількість реквізитів в формі документа, що заповнюються вручну, з довідника банку та з довідника клієнта, відповідно;

$\omega_1$  – контроль правильності заповнення реквізитів на стороні клієнта;

$U_1$  – доробка документа при виявленні помилок контролем  $\omega_1$ ;

$A_4$  – електронно-цифрове підписування документа;

$A_5$  – передача документа каналом зв'язку;

$\omega_2$  – контроль реквізитів та підпису документа на стороні банку;

$U_2$  – повідомлення клієнта про відмову виконання запиту;

$A_6$  – завантаження документа в автоматизовану банківську систему;

$\omega_3$  – контроль документа в автоматизованій банківській системі;

$B = U_2 \left\{ U_1 \right\}_{\omega_1} A_4 A_5$  – позначення для спрощеного запису алгоритму

(6.52).

Схему алгоритму проведення платіжного доручення зображено на рис. 6.5.

При виконанні алгоритму (6.52) можливі такі стани [3]:

«0» – помилки відсутні;

«1» – форматна помилка;

«2» – операційна помилка;

«3» – помилка електронно-цифрового підпису.

Класифікацію можливих помилок наведено в додатку Б.

На етапі проектування системи скільки часу потрібно для обробки запиту клієнта [3]. Спрогнозуємо цей показник при нечітких даних про ймовірно-часові характеристики операторів та логічних умов алгоритму, які наведені в табл. 6.11. Початкові дані отримано в результаті опитування проектувальників системи. Для лінгвістичної оцінки ймовірно-часових характеристик використовується термножина {Низький, Середній, Вище Середнього, Високий}. Характеристики надійності операторів та логічних умов задані трикутними функціями належності.  $\alpha$ -форми нечітких чисел наведено в додатку В.

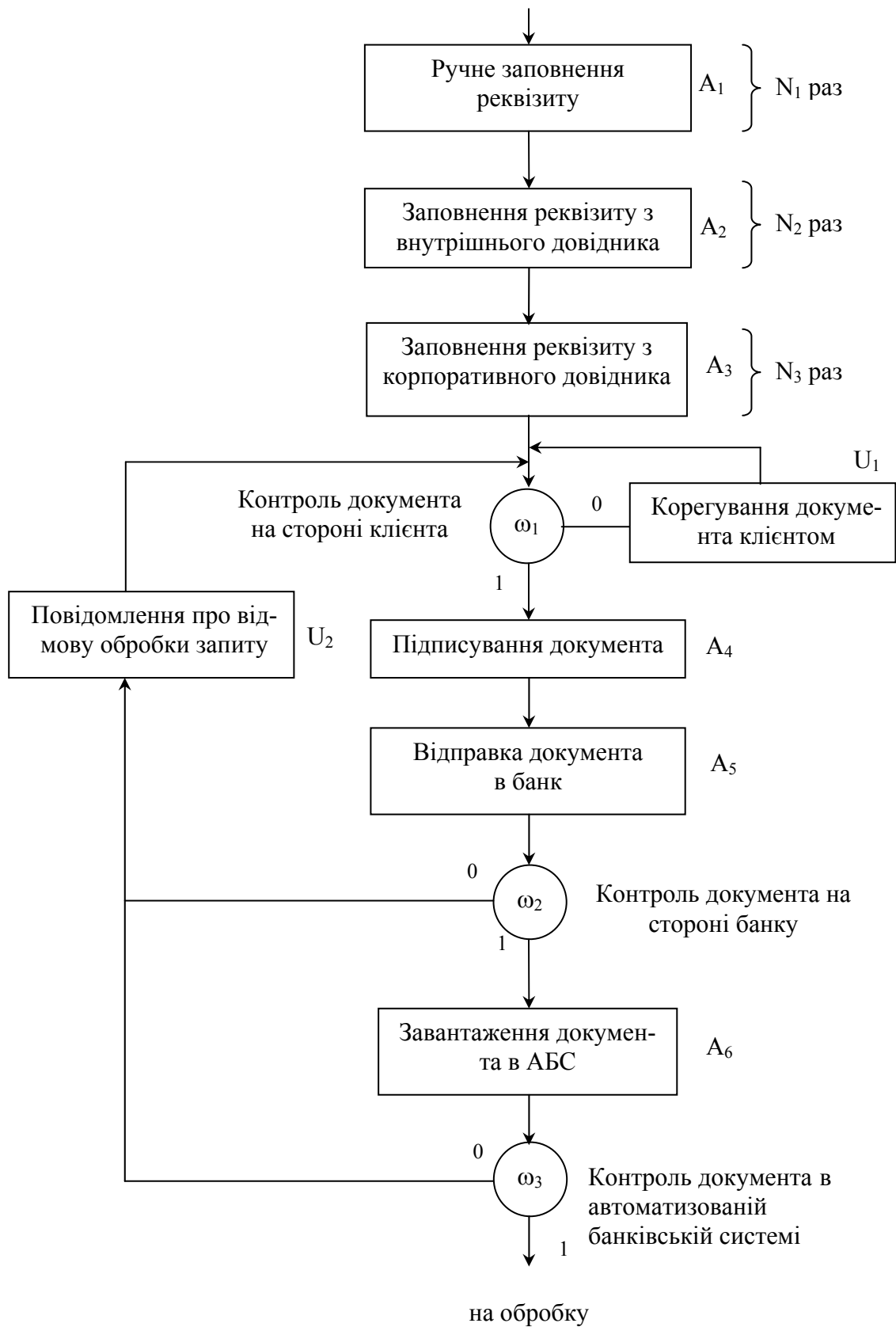


Рис. 6.5. Схема алгоритму проведення платіжного доручення



Таблиця 6.11

Характеристики надійності алгоритму (6.52)

Елемент	Характеристика	Нижня межа	Верхня межа	Лінгвістична оцінка
$A_1$	$\tilde{p}_{A_1}^1$	0,995	0,999	Вище середнього
	$\tilde{p}_{A_1}^{0_1}$	0	0,006	Середній
	$\tilde{p}_{A_1}^{0_2}$	0	0,0003	Середній
	$\tilde{t}_{A_1}$	4	6	Високий
	$N_1$	6	6	–
$A_2$	$\tilde{p}_{A_2}^1$	0,9997	0,9999	Високий
	$\tilde{p}_{A_2}^{0_1}$	0	0,0008	Низький
	$\tilde{p}_{A_2}^{0_2}$	0	0,0004	Низький
	$\tilde{t}_{A_2}$	2	4	Середній
	$N_2$	4	4	–
$A_3$	$\tilde{p}_{A_3}^1$	0,9996	0,9999	Високий
	$\tilde{p}_{A_3}^{0_1}$	0	0,0009	Низький
	$\tilde{p}_{A_3}^{0_2}$	0	0,0003	Низький
	$\tilde{t}_{A_3}$	1,5	3	Середній
	$N_3$	4	4	–
$A_4$	$\tilde{p}_{A_4}^1$	0,99997	1	Вище середнього
	$\tilde{p}_{A_4}^{0_3}$	0	0,00005	Низький
	$\tilde{t}_{A_4}$	3	30	Середній
$A_5$	$\tilde{p}_{A_5}^1$	1	1	–
	$\tilde{t}_{A_5}$	15	150	Вище середнього
$A_6$	$\tilde{p}_{A_6}^1$	1	1	–
	$\tilde{t}_{A_6}$	5	60	Середній

Продовження таблиці 6.11

Елемент	Характеристика	Нижня межа	Верхня межа	Лінгвістична оцінка
$U_1$	$\tilde{v}_{U_1}^{1_1}$	0,9	0,93	Низький
	$\tilde{v}_{U_1}^{1_2}$	0,92	0,95	Середній
	$\tilde{t}_{U_1}$	10	35	Високий
$U_2$	$\tilde{v}_{U_2}^1$	1	1	– Вище середнього
	$\tilde{t}_{U_2}$	15	150	
$\omega_1$	$\tilde{k}_{\omega_1}^{11}$	1	1	–
	$\tilde{k}_{\omega_1}^{00_1}$	0,88	0,96	Високий
	$\tilde{k}_{\omega_1}^{00_2}$	0,93	0,97	Вище середнього
	$\tilde{t}_{\omega_1}$	0,5	1	Низький
$\omega_2$	$\tilde{k}_{\omega_2}^{11}$	1	1	–
	$\tilde{k}_{\omega_2}^{00_1}$	0,88	0,96	Високий
	$\tilde{k}_{\omega_2}^{00_2}$	0,93	0,97	Вище середнього
	$\tilde{k}_{\omega_2}^{00_3}$	1	1	–
	$\tilde{t}_{\omega_2}$	0,5	1,2	Низький
$\omega_3$	$\tilde{k}_{\omega_3}^{11}$	1	1	–
	$\tilde{k}_{\omega_3}^{00_1}$	0,82	0,87	Високий
	$\tilde{k}_{\omega_3}^{00_2}$	0,9	0,95	Високий
	$\tilde{t}_{\omega_3}$	1	1,5	Низький

Алгоритм (6.52) містить типові алгоритмічні структури «послідовна», « $\omega$ -доробка», «робота-контроль-доробка» та «багаторазова робота». Застосовуючи моделі надійності цих алгоритмічних структур, отримуємо  $\alpha$ -форми часу виконання алгоритму (6.52)

$$\tilde{t}_Y = (76,4, 348,2)_0 \cup (115,6, 319,1)_{0,25} \cup (154,9, 290,2)_{0,5} \cup (194,3, 261,8)_{0,75} \cup (233,7, 233,7)_1.$$

Отримане нечітке число  $\tilde{t}_Y$  зображено на рис. 6.6. Його можна інтерпретувати таким чином

$$\tilde{t}_Y = \langle 76,4, 348,2, \text{середній} \rangle.$$

Покроковий розрахунок надійності АП (6.52) наведено в додатку Г.

На рис. 6.7 показано реальний ймовірнісний розподіл часу реакції системи «клієнт-банк» та прогнозовану нечітку оцінку  $\tilde{t}_Y$ . Реальне розподілення отримано при навантаженому тестуванні системи протягом трьох місяців 2005 р. Математичне очікування за цим розподілом складає 224 с. Після дефазифікації нечіткого числа  $\tilde{t}_Y$  за методом центра тяжіння отримуємо тривалість середнього часу 220 с. З рис. 6.7 видно, що отримана оцінка нечіткого часу  $\tilde{t}_Y$  практично збігається з експериментальними даними. Розбіжність між результатами дефазифікації та математичним очікуванням складає 4 с. Це підтверджує достовірність нечітких моделей надійності алгоритмічних структур, що запропоновані в підрозділі 6.4.

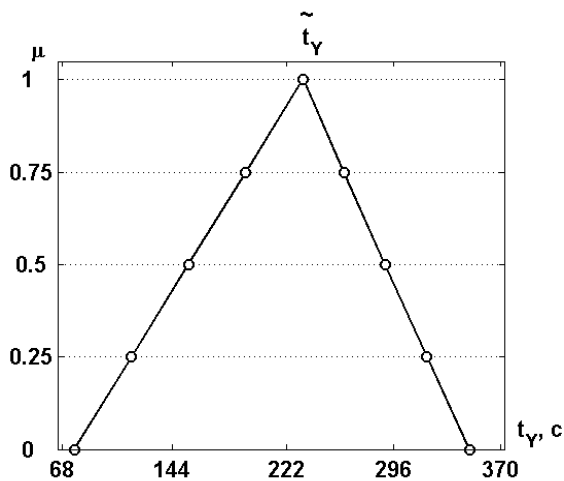


Рис. 6.6. Прогнозований час виконання алгоритму (6.52)

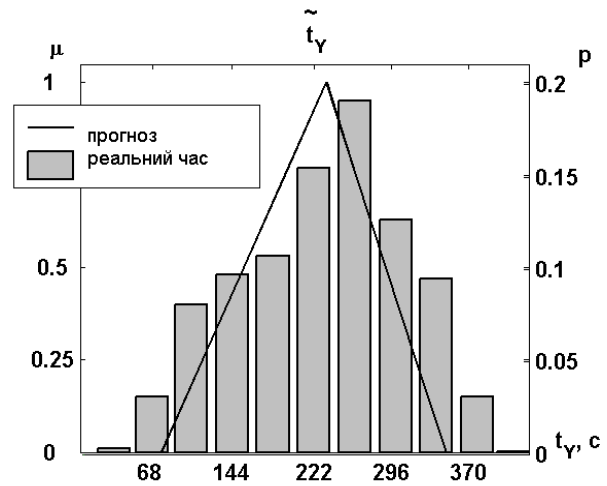


Рис. 6.7. Порівняння реального розподілу часу з прогнозованим

Порівняємо результати чіткого та нечіткого прогнозування. Для цього:

1) використаємо координати максимумів трикутних функцій належностей з додатку В як значення ймовірно-часових характери-

стик операторів та логічних умов;

2) спрогнозуємо час  $t_Y$  за допомогою алгоритму аналізу надійності АП, що наведено [79].

В результаті отримуємо, що чіткий прогноз середнього часу проведення платіжних доручень складає 234 с, що на 10 с більше реального.

Таким чином, на етапі проектування алгоритму (6.52) нечіткі моделі надійності дозволяють підвищити достовірність прогнозу за рахунок використання лінгвістичних оцінок.

Проведемо аналіз чутливості часу обробки запиту  $\tilde{t}_Y$  до лінгвістичного значення нечіткого часу передачі платіжного доручення каналом зв'язку. Для цього будемо змінювати координату максимуму  $\hat{t}_{A_5}$  трикутної функції належності нечіткого числа  $\tilde{t}_{A_5}$  в діапазоні [15, 150]. Цей діапазон відповідає зміні лінгвістичної оцінки від «низької» до «високої».

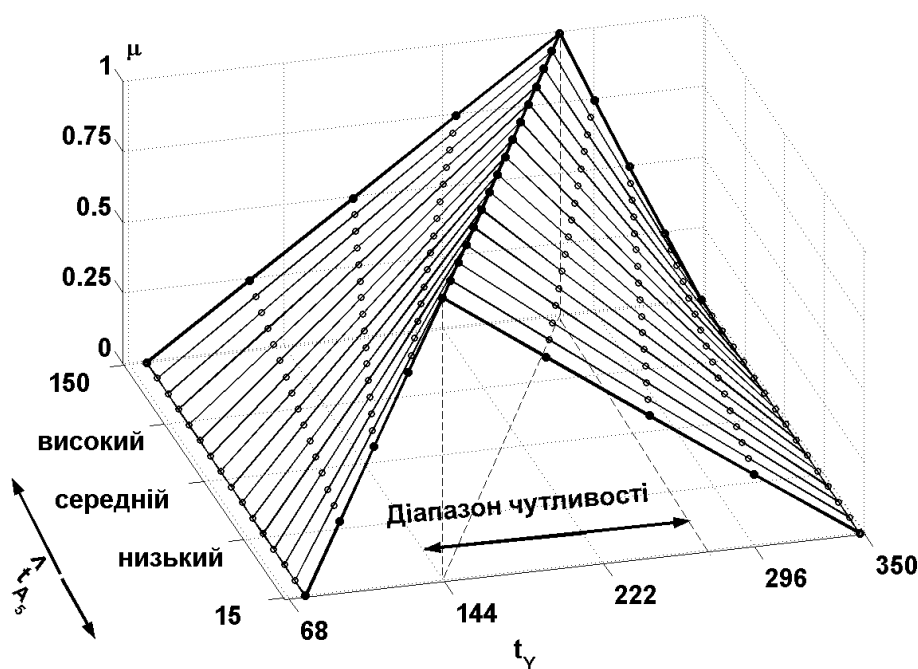


Рис. 6.8. Аналіз чутливості нечіткого часу  $\tilde{t}_Y$  до параметра  $\hat{t}_{A_5}$

В результаті аналізу чутливості встановлено, що лінгвістична оцінка часу виконання алгоритму  $\tilde{t}_Y$  змінюється від «Нижче Середнього» до «Вище Середнього». Графік чутливості нечіткого часу  $\tilde{t}_Y$  до варіації параметра  $\hat{t}_{A_5}$  зображено на рис. 6.8.

## 7. НЕЧІТКА ОПТИМІЗАЦІЯ НАДІЙНОСТІ АЛГОРИТМІЧНИХ ПРОЦЕСІВ

Пропонуються постановки та методи розв'язання задач оптимізації надійності багатовимірних АП для випадків нечіткої мети і обмежень задачі оптимізації та нечітких показників надійності альтернатив. Нечітка оптимізація здійснюється ГА, в якому хромосоми оцінюються за принципом Беллмана–Заде з нечіткої теорії прийняття рішень та обираються в нову популяцію за схемою турнірної селекції.

### 7.1. Концепція нечіткої оптимізації надійності

Керовані змінні в задачах нечіткої оптимізації надійності АП будуть такі ж, як і в чіткому випадку. Таким чином, задачами нечіткої оптимізації надійності АП можуть бути розстановка контрольних точок, вибір кратностей контролів, вибір реалізацій операторів та логічних умов та інші задачі структурної, параметричної та структурно-параметричної оптимізації АП (див. табл. 2.1).

Нечітку оптимізацію надійності АП пропонується здійснювати на основі ГА з розділу 5. При цьому операції схрещування та мутації, які генерують варіанти АП, залишаються без змін. Відбір хромосом в нову популяцію здійснимо за запропонованою новою схемою турнірної селекції, яка демонструє добру динаміку оптимізації і, найголовніше, не потребує штрафних коефіцієнтів для врахування обмежень. Єдиним елементом ГА, який потрібно розробити для нечіткої оптимізації надійності АП, є *фітнес функція*, тобто модуль порівняння хромосом за нечіткими значеннями показників надійності та (або) за нечітких обмежень задачі оптимізації. Нижче розглядається як реалізувати порівняння альтернатив в таких нечітких умовах.

Для врахування нечіткості при оптимізації надійності АП спробуємо поєднати ідеї багатокритеріального прийняття рішень за схемою Беллмана–Заде [4, 121] та цільового програмування з пункту 3.4.2. Вважатимемо, що мета оптимізації задана вектором  $\tilde{G} = (P^*, C^*, T^*, I^*, S^*, q_j)$  бажаних значень показників  $p^1, C, T, I, S$  та  $p^{0j}, j = \overline{1, m}$  надійності АП. Будемо вважати відомими ступені задоволення мети оптимізації  $\mu_{\tilde{p}^*}(\mathbf{X}), \mu_{\tilde{C}^*}(\mathbf{X}), \mu_{\tilde{T}^*}(\mathbf{X}), \mu_{\tilde{I}^*}(\mathbf{X}), \mu_{\tilde{S}^*}(\mathbf{X})$  та  $\mu_{\tilde{q}_j}(\mathbf{X})$  варіантом АП, який задано вектором керованих змінних  $\mathbf{X}, j = \overline{1, m}$ . Згідно з принципом Беллмана–Заде ступінь виконання мети оптимізації за усіма критеріями одночасно розрахуємо таким чином:

$$\mu_{\tilde{G}}(\mathbf{X}) = \mu_{\tilde{P}^*}(\mathbf{X}) \wedge \mu_{\tilde{C}^*}(\mathbf{X}) \wedge \mu_{\tilde{T}^*}(\mathbf{X}) \wedge \mu_{\tilde{I}^*}(\mathbf{X}) \wedge \mu_{\tilde{S}^*}(\mathbf{X}) \wedge \mu_{\tilde{q}_j}(\mathbf{X}), \quad j = \overline{1, m}, \quad (7.1)$$

де  $\wedge$  – операція мінімуму.

За (7.1) та за відомими функціями належностей бажаних значень показників надійності АП нечітку мету оптимізації можна представити нечітким відношенням  $\tilde{G}$  на декартовому добутку критеріїв. Підставляючи в функцію належності цього нечіткого відношення показники надійності варіанта  $\mathbf{X}$ , отримуємо  $\mu_{\tilde{G}}(\mathbf{X})$  – ступінь належності АП до нечіткої мети оптимізації. Як представити нечітку мету у вигляді нечіткого відношення розглянемо на такому прикладі.

*Приклад 7.1.* Розглядається задача оптимізації АП в такій постановці:

$$\begin{cases} p^1(\mathbf{X}) = \tilde{P}^* \\ C(\mathbf{X}) = \tilde{C}^* \end{cases}$$

Графіки функцій належності нечітких чисел  $\tilde{P}^*$  та  $\tilde{C}^*$  зображені на рис. 7.1.

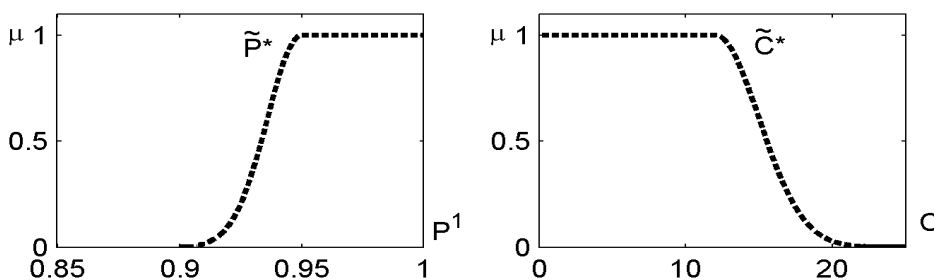


Рис. 7.1. Нечіткі числа з прикладу 7.1

З (7.1) нечітку мету оптимізації можна зобразити еквівалентним нечітким відношенням на  $P \times C$  (рис. 7.2). Функція належності нечіткого відношення розраховується таким чином:

$$\mu_{\tilde{G}}(P, C) = \min(\mu_{\tilde{P}}(P), \mu_{\tilde{C}}(C)), \quad P \in \text{supp}(\tilde{P}), \quad C \in \text{supp}(\tilde{C}).$$

На рис. 7.3 незалита частина поверхні нечіткого відношення показує область повного задоволення мети оптимізації. Найтемніша частина поверхні відповідає області недопустимих розв'язків. Ступінь виконання нечіткого відношення тотожний ступеню виконання нечіткої мети оптимізації АП.  $\boxtimes$

За ідеєю нечіткої теорії прийняття рішень [180] за нерівноважливих критеріїв формула (7.1) перетворюється на таку:

$$\mu_{\tilde{G}}(\mathbf{X}) = \mu_{\tilde{P}^*}(\mathbf{X})^{w_P} \wedge \mu_{\tilde{C}^*}(\mathbf{X})^{w_C} \wedge \mu_{\tilde{T}^*}(\mathbf{X})^{w_T} \wedge \mu_{\tilde{I}^*}(\mathbf{X})^{w_I} \wedge \mu_{\tilde{S}^*}(\mathbf{X})^{w_S} \wedge \mu_{\tilde{q}_j}(\mathbf{X})^{w_{qj}}, \quad j = \overline{1, m} \quad (7.2)$$

де  $w(\cdot) > 0$  – коефіцієнти важливості відповідних критеріїв, які знаходять різними методами обробки експертних думок, наприклад, за парними порівняннями Сааті.

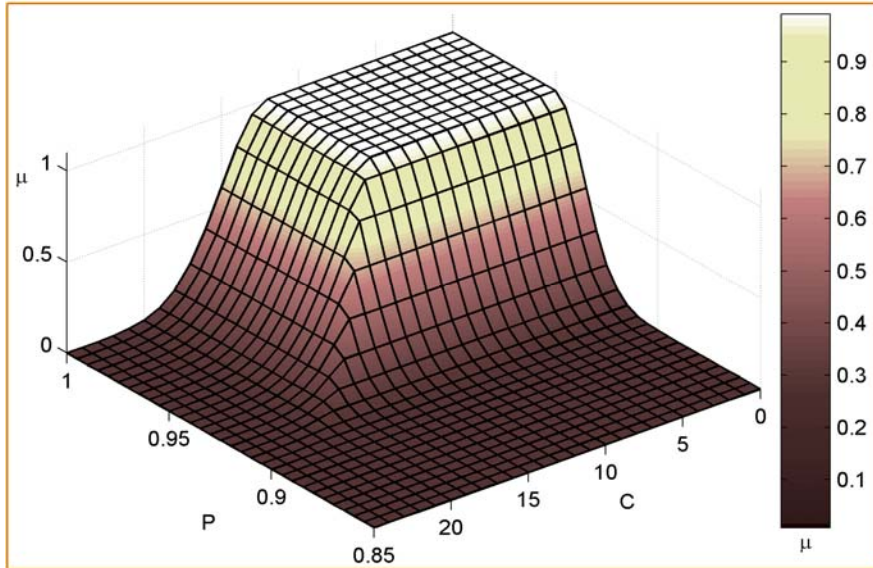


Рис. 7.2. До прикладу 7.1: нечітка мета оптимізації в вигляді нечіткого відношення

За (7.2) та за відомими нечіткими бажаними значеннями показників надійності АП мету оптимізації можна представити нечітким відношенням  $\tilde{G}$  на декартовому добутку критеріїв. Як це зробити розглянемо на такому прикладі.

*Приклад 7.2.* За даними попереднього прикладу представимо нечітку мету оптимізації нечітким відношенням за умови нерівноважливих критеріїв.

За коефіцієнтів важливостей  $w_P$  та  $w_C$  критеріїв  $P$  та  $C$  функція належності нечіткого відношення розраховується таким чином:

$$\mu_{\tilde{G}}(P, C) = \min(\mu_{\tilde{P}}(P)^{w_P}, \mu_{\tilde{C}}(C)^{w_C}), \quad P \in \text{supp}(\tilde{P}), C \in \text{supp}(\tilde{C}).$$

Нечітке відношення для різних значень коефіцієнтів важливості зображено на рис. 7.3. ☒

Таким чином, запропоновано прості реалізації усіх етапів нечіткої оптимізації за винятком одного – розрахунку ступеня виконання варіантом АП мети оптимізації. Нижче розробляються правила розрахунку цих ступенів для кожного з трьох вищеописаних випадків нечіткої оптимізації.

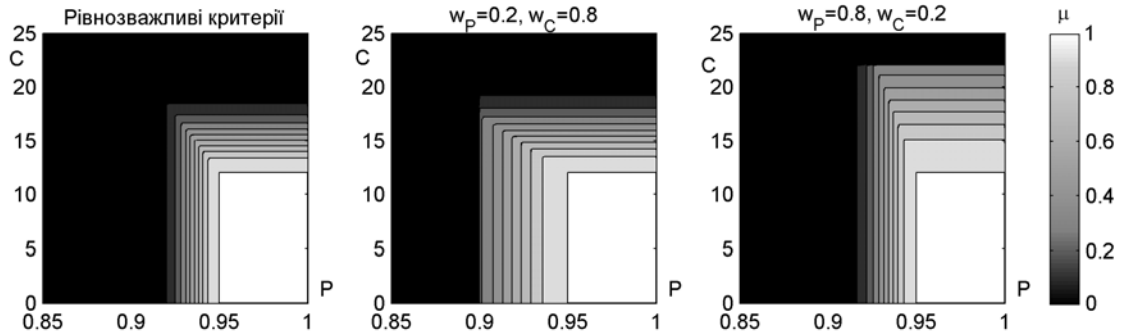


Рис. 7.3. До прикладу 7.2: нечітка мета оптимізації в вигляді нечіткого відношення при різних коефіцієнтах важливості критеріїв

## 7.2. Оптимізації за нечіткої мети

Нечітку мету оптимізації сформулюємо у вигляді вектора бажаних нечітких значень показників надійності АП

$$\tilde{G} = (\tilde{P}^*, \tilde{C}^*, \tilde{T}^*, \tilde{q}_j, \tilde{I}^*, \tilde{S}^*), \quad j = \overline{1, m}.$$

Ці значення назвемо *цільовими нечіткими числами*. Вони визначають нечітку область допустимих розв'язків задачі оптимізації АП. Ядра цільових нечітких чисел визначають чітку область розв'язків, в якій обмеження виконано повною мірою. Відповідно до (1.5) ядра цільових нечітких чисел є такими:

$$\left. \begin{aligned} \text{core}(\tilde{P}^*) &= [P^*, 1]; \\ \text{core}(\tilde{C}^*) &= [0, C^*]; \\ \text{core}(\tilde{T}^*) &= [0, T^*]; \\ \text{core}(\tilde{I}^*) &= [0, I^*]; \\ \text{core}(\tilde{S}^*) &= [0, S^*]; \\ \text{core}(\tilde{p}^{0j}) &= [0, q_j], \quad j = \overline{1, m}. \end{aligned} \right\} \quad (7.3)$$

Узагальнюючи задачу (3.11), оптимізацію АП при чітких початкових даних та нечітких цілях поставимо як пошук вектора  $\mathbf{X}$ , при якому:

$$\left\{ \begin{aligned} p^1(\mathbf{X}) &= \tilde{P}^*; \\ p^{0j}(\mathbf{X}) &= \tilde{q}_j, \quad j = \overline{1, m}; \\ I(\mathbf{X}) &= \tilde{I}^*; \\ C(\mathbf{X}) &= \tilde{C}^*; \\ T(\mathbf{X}) &= \tilde{T}^*; \\ S(\mathbf{X}) &= \tilde{S}^*. \end{aligned} \right. \quad (7.4)$$



Ступені досягнення варіантом  $\mathbf{X}$  мети оптимізації  $\tilde{G}$  за кожним нечітким критерієм позначимо через  $\mu_{\tilde{P}^*}(\mathbf{X})$ ,  $\mu_{\tilde{C}^*}(\mathbf{X})$ ,  $\mu_{\tilde{T}^*}(\mathbf{X})$ ,  $\mu_{\tilde{I}^*}(\mathbf{X})$ ,  $\mu_{S^*}(\mathbf{X})$  та  $\mu_{\tilde{q}_j}(\mathbf{X})$ ,  $j = \overline{1, m}$ . Ступені належності знаходимо підстановкою чітких чисел  $P(\mathbf{X})$ ,  $C(\mathbf{X})$ ,  $T(\mathbf{X})$ ,  $I(\mathbf{X})$ ,  $S(\mathbf{X})$  та  $q_j(\mathbf{X})$  в формули функції належності цільових нечітких чисел  $\tilde{P}^*$ ,  $\tilde{C}^*$ ,  $\tilde{T}^*$ ,  $\tilde{I}^*$ ,  $\tilde{S}^*$  та  $\tilde{q}_j$  ( $j = \overline{1, m}$ ). Якщо для варіанта  $\mathbf{X}$  якийсь показник АП знаходиться за межами носія цільових нечітких чисел, тоді ступень виконання відповідного критерію дорівнюватиме 0.

Серед двох варіантів  $\mathbf{X}_1$  та  $\mathbf{X}_2$  кращим буде АП з більшим значенням ступеня належності (7.8). Тобто, якщо  $\mu_{\tilde{G}}(\mathbf{X}_1) > \mu_{\tilde{G}}(\mathbf{X}_2)$ , тоді обираємо  $\mathbf{X}_1$ . Відповідно, якщо  $\mu_{\tilde{G}}(\mathbf{X}_1) < \mu_{\tilde{G}}(\mathbf{X}_2)$ , тоді обираємо  $\mathbf{X}_2$ . При рівних ступенях належності  $\mu_{\tilde{G}}(\mathbf{X}_1) = \mu_{\tilde{G}}(\mathbf{X}_2)$  рішення приймаємо за найважливішим критерієм (з максимальним значенням вагового коефіцієнту). Якщо  $\mathbf{X}_1$  та  $\mathbf{X}_2$  є неприпустимими, тобто якщо  $\mu_{\tilde{G}}(\mathbf{X}_1) = \mu_{\tilde{G}}(\mathbf{X}_2) = 0$ , тоді обираємо варіант з меншим числом порушень цілей оптимізації. При рівній кількості порушень обираємо варіант з кращим значенням найважливішого критерію. Описану вище послідовність дій сформулюємо у вигляді такого алгоритму вибору кращого розв'язку ( $\mathbf{X}$ ) з альтернатив  $\mathbf{X}_1$  та  $\mathbf{X}_2$ :

```

If  $\mu_{\tilde{G}}(\mathbf{X}_1) > \mu_{\tilde{G}}(\mathbf{X}_2)$  then  $\mathbf{X} := \mathbf{X}_1$ ;
  elseif  $\mu_{\tilde{G}}(\mathbf{X}_2) > \mu_{\tilde{G}}(\mathbf{X}_1)$  then  $\mathbf{X} := \mathbf{X}_2$ ;
    elseif  $\mu_{\tilde{G}}(\mathbf{X}_1) > 0$  then <обрати альтернативу за кращим
      значенням найважливішого критерію>;
      elseif  $E(\mathbf{X}_1) < E(\mathbf{X}_2)$  then  $\mathbf{X} := \mathbf{X}_1$ ;
        elseif  $E(\mathbf{X}_1) > E(\mathbf{X}_2)$  then  $\mathbf{X} := \mathbf{X}_2$ ;
    else <обрати альтернативу за кращим значенням важливішого
      критерію>;
  end
  %E – кількість порушених цілей оптимізації.

```

Якщо в задачі оптимізації якийсь з критеріїв не застосовано, тоді відповідний ступінь належності прирівнюємо до 1, тобто вважатимемо, що обмеження виконано. Якщо обмеження за якимось критерієм задано нерівністю з чітким пороговим значенням, тоді представимо його нечітким критерієм з ступінчатою функцією належності. Наприклад, обмеження (1.3) задамо нечітким числом, функція належності

якого приймає нульові значення для  $C(\mathbf{X}) > 9$ , та дорівнює 1 коли  $C(\mathbf{X}) \leq 9$ .

Якщо в задачі оптимізації важливість критеріїв однакова, тоді особа, що відповідає за прийняття рішення, призначає найважливіший критерій, за яким обирається розв'язок коли  $\mu_{\tilde{G}}(\mathbf{X}_1) = \mu_{\tilde{G}}(\mathbf{X}_2)$ . В цьому випадку його можна інтерпретувати як критерій нечіткої оптимальності, а решту показників АП як нечіткі обмеження задачі оптимізації.

*Приклад 7.3.* Розглядається задача вибору варіантів реалізації операторів АП за нечіткої мети оптимізації. АП складається з послідовного виконання двох операторів –  $Y = A_1 A_2$ , кожен з яких можна реалізувати двома способами:  $A_1 \in \{A_{1\_1}, A_{1\_2}\}$  та  $A_2 \in \{A_{2\_1}, A_{2\_2}\}$ . Характеристики надійності операторів зведені в табл. 7.1.

Таблиця 7.1

Характеристики надійності операторів з прикладу 7.3

Оператори	Характеристики		
	$p^1$	$C$	$T$
$A_{1\_1}$	0,96	4	3
$A_{1\_2}$	0,99	9	5
$A_{2\_1}$	0,975	5	6
$A_{2\_2}$	0,97	6	5

Задача оптимізації ставиться як пошук вектора  $\mathbf{X}$  реалізацій операторів, який забезпечує:

$$\begin{cases} p^1(\mathbf{X}) \geq \tilde{P}^*; \\ C(\mathbf{X}) \leq \tilde{C}^*; \\ T(\mathbf{X}) \leq \tilde{T}^*, \end{cases}$$

де цільові нечіткі числа задані такими функціями належності:

$$\mu_{\tilde{P}^*}(p) = \begin{cases} 0, & \text{якщо } p < 0,9 \\ \exp\left(-\frac{(p - 0,95)^2}{2 \cdot 0,014^2}\right), & \text{якщо } p \in [0,9; 0,95]; \\ 1, & \text{якщо } p > 0,95 \end{cases}$$

$$\mu_{\tilde{C}^*}(C) = \begin{cases} 0, & \text{якщо } C > 22 \\ \exp\left(-\frac{(C-12)^2}{2 \cdot 3^2}\right), & \text{якщо } C \in [12, 22]; \\ 1, & \text{якщо } C < 12 \end{cases}$$

$$\mu_{\tilde{T}^*}(T) = \begin{cases} 0, & \text{якщо } T > 12 \\ \exp\left(-\frac{(T-9)^2}{2 \cdot 0.8^2}\right), & \text{якщо } T \in [9, 12]. \\ 1, & \text{якщо } T < 9 \end{cases}$$

Задачу розв'яжемо повним перебором, тому що є лише такі чотири можливі варіанти значень керованих змінних:

$\mathbf{X}=(1, 1)$ , якій відповідає АП  $Y = A_{1\_1}A_{2\_1}$ ;

$\mathbf{X}=(1, 2)$ , якій відповідає АП  $Y = A_{1\_1}A_{2\_2}$ ;

$\mathbf{X}=(2, 1)$ , якій відповідає АП  $Y = A_{1\_2}A_{2\_1}$ ;

$\mathbf{X}=(2, 2)$ , якій відповідає АП  $Y = A_{1\_2}A_{2\_2}$ .

Показники надійності кожного варіанта АП розрахуємо за моделями послідовної структури [79] таким чином:

$$p^1(\mathbf{X} = (1, 1)) = 0,96 \cdot 0,975 = 0,936;$$

$$C(\mathbf{X} = (1, 1)) = 4 + 5 = 9;$$

$$T(\mathbf{X} = (1, 1)) = 3 + 5 = 8;$$

$$p^1(\mathbf{X} = (1, 2)) = 0,96 \cdot 0,97 = 0,931;$$

$$C(\mathbf{X} = (1, 2)) = 4 + 6 = 10;$$

$$T(\mathbf{X} = (1, 2)) = 3 + 6 = 9;$$

$$p^1(\mathbf{X} = (2, 1)) = 0,99 \cdot 0,975 = 0,965;$$

$$C(\mathbf{X} = (2, 1)) = 9 + 5 = 14;$$

$$T(\mathbf{X} = (2, 1)) = 3 + 6 = 9;$$

$$p^1(\mathbf{X} = (2, 2)) = 0,99 \cdot 0,97 = 0,96;$$

$$C(\mathbf{X} = (2, 2)) = 9 + 6 = 15;$$

$$T(\mathbf{X} = (2, 2)) = 5 + 5 = 10.$$

Підставляючи розраховані значення в функції належності цільових нечітких чисел отримуємо ступені належності розв'язків до мети оптимізації. Процес розрахунку цих ступенів належності зображено на рис. 7.4.

Підставляючи в (7.1) ступені належності з рис. 7.5, отримуємо:

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,607 \wedge 1 \wedge 1 = 0,607;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0,406 \wedge 1 \wedge 1 = 0,406;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 1 \wedge 0,801 \wedge 0,044 = 0,044;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 1 \wedge 0,607 \wedge 0,458 = 0,458.$$

За цими ступенями належності альтернативи впорядкуємо таким чином:

1-ий ранг для  $\mathbf{X}=(1, 1)$ ;

2-ий ранг для  $\mathbf{X}=(2, 2)$ ;

3-ій ранг для  $\mathbf{X}=(1, 2)$ ;

4-ий ранг для  $\mathbf{X}=(2, 1)$ .

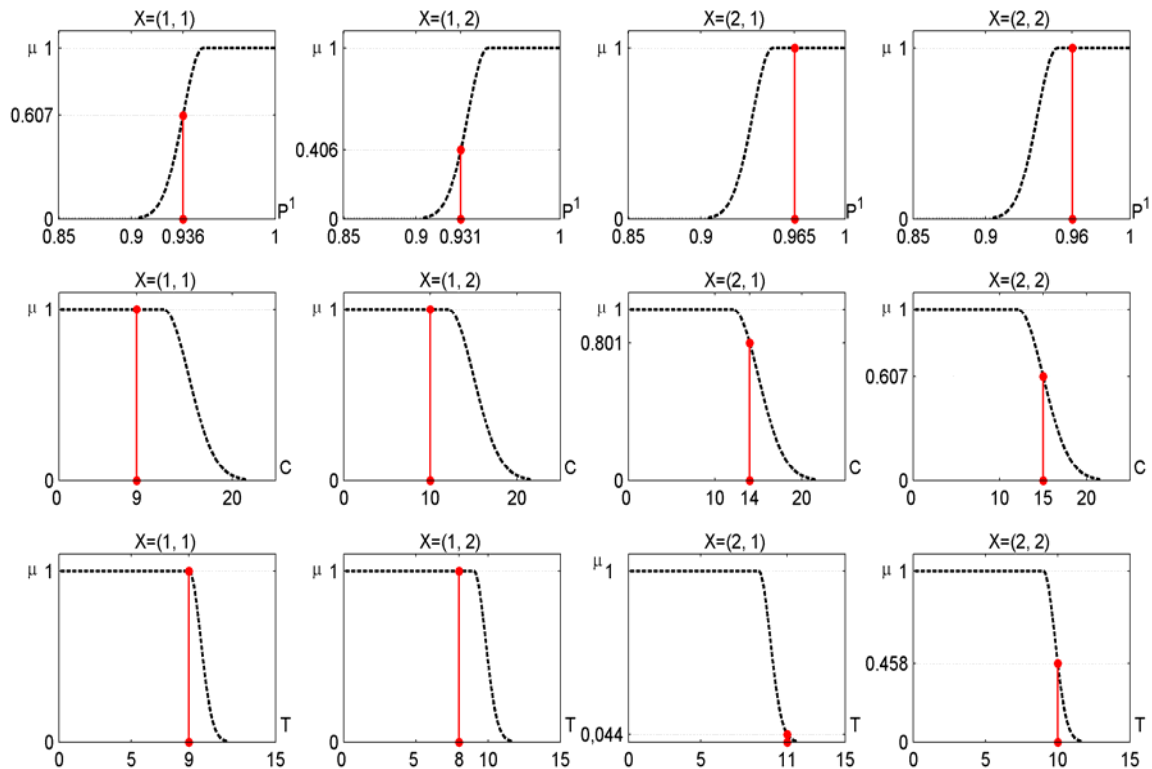


Рис. 7.4. Виконання розв'язками нечіткої мети оптимізації в прикладі 7.3

Проранжуємо альтернативи для нерівноважливих критеріїв. Нехай коефіцієнти важливості критеріїв є такими:  $w_P=0,7$ ,  $w_C=0,1$  та  $w_T=0,2$ . Тоді, за (7.2) отримуємо:

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,607^{0,7} \wedge 1^{0,1} \wedge 1^{0,2} = 0,705;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0,406^{0,7} \wedge 1^{0,1} \wedge 1^{0,2} = 0,532;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 1^{0.7} \wedge 0,801^{0.1} \wedge 0,044^{0.2} = 0,535;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 1^{0.7} \wedge 0,607^{0.1} \wedge 0,458^{0.2} = 0,855.$$

Коефіцієнти важливості змінили відношення порядку для двох пар альтернатив. Тепер альтернативи впорядковані таким чином:

- 1-ий ранг для  $\mathbf{X}=(2, 2)$ ;
- 2-ий ранг для  $\mathbf{X}=(1, 1)$ ;
- 3-ій ранг для  $\mathbf{X}=(2, 1)$ ;
- 4-ий ранг для  $\mathbf{X}=(1, 2)$ .  $\square$

### 7.3. Оптимізація за нечітких початкових даних

Розглянемо випадок коли значення характеристик надійності елементів АП задані нечіткими числами. За нечіткими моделями надійності АП для розв'язку  $\mathbf{X}$  отримуємо нечіткі значення  $\tilde{p}^1(\mathbf{X})$ ,  $\tilde{C}(\mathbf{X})$ ,  $\tilde{T}(\mathbf{X})$ ,  $\tilde{q}_j(\mathbf{X})$  ( $j = \overline{1, m}$ ),  $\tilde{I}(\mathbf{X})$ . Метою цього підрозділу є знаходження відповідей на такі питання:

- як визначити ступінь виконання нечіткими показниками надійності чітких цілей оптимізації?
- як визначити найкращий розв'язок за нечіткими показниками надійності?

Задачу оптимізації АП при нечітких початкових даних та чітких цілях оптимізації поставимо таким чином: знайти вектор  $\mathbf{X}$ , що забезпечує:

$$\left\{ \begin{array}{l} \tilde{p}^1(\mathbf{X}) = P^*; \\ \overline{\text{supp}}(\tilde{p}^1(\mathbf{X})) \geq \underline{P}^*; \\ \tilde{p}^{0j}(\mathbf{X}) = q_j, \quad j = \overline{1, m}; \\ \overline{\text{supp}}(\tilde{p}^{0j}(\mathbf{X})) \leq \overline{q}_j; \\ \tilde{I}(\mathbf{X}) = I^*; \\ \overline{\text{supp}}(\tilde{I}(\mathbf{X})) \leq \overline{I}^*; \\ \tilde{C}(\mathbf{X}) = C^*; \\ \overline{\text{supp}}(\tilde{C}(\mathbf{X})) \leq \overline{C}^*; \\ \tilde{T}(\mathbf{X}) = T^*; \\ \overline{\text{supp}}(\tilde{T}(\mathbf{X})) \leq \overline{T}^*; \\ S(\mathbf{X}) \leq S^*, \end{array} \right. \quad (7.5)$$

де  $\underline{p}^*$  – мінімальне граничне значення  $p^1(\mathbf{X})$ ;

$\bar{q}_j$  – максимальне граничне значення  $p^{0j}(\mathbf{X})$ ;

$\bar{I}^*$  – максимальне граничне значення  $I(\mathbf{X})$ ;

$\bar{C}^*$  – максимальне граничне значення  $C(\mathbf{X})$ ;

$\bar{T}^*$  – максимальне граничне значення  $T(\mathbf{X})$ ;

$\underline{\text{supp}}$  та  $\overline{\text{supp}}$  – мінімальне та максимальне значення носія нечіткого числа.

В (7.5) обмеження з  $S(\mathbf{X})$  записано в чіткій формі, тому що кількість контрольних операцій в АП, що синтезується в результаті оптимізації, є чітким числом. Введення в (7.5) граничних значень ( $\underline{P}^*$ ,  $\bar{q}_j$ ,  $\bar{I}^*$ ,  $\bar{C}^*$ ,  $\bar{T}^*$ ) гарантує, що в будь-якому випадку показники надійності синтезованого АП не вийдуть за встановлені межі. Іншими словами, граничні значення які входять в чіткі обмеження–нерівності визначають чітку область допустимих розв’язків. Їх можна розглядати як аналогі обмежень на хвості розподілень випадкових величин при оптимізації в умовах ймовірнісної невизначеності [8]. Якщо особу, яка відповідає за прийняття рішення, не лякає можливість виходу деякого показника АП за дозволіні межі, тоді вилучимо відповідні граничні значення з постановки задачі оптимізації.

Ступені досягнення варіантом  $\mathbf{X}$  мети оптимізації за кожним критерієм розрахуємо таким чином:

$$\mu_{\tilde{P}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \underline{\text{supp}}(\tilde{p}^1(\mathbf{X})) < \underline{P}^* \\ 0, & \text{якщо } \underline{\text{core}}(\tilde{p}^1(\mathbf{X})) < \underline{P}^* \\ 0, & \text{якщо } \text{centroid}(\tilde{p}^1(\mathbf{X})) < \underline{P}^*; \\ 1 - \mu_{\tilde{p}^1(\mathbf{X})}(\underline{P}^*), & \text{інакше} \end{cases} \quad (7.6)$$

$$\mu_{\tilde{C}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{C}(\mathbf{X})) > \bar{C}^* \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{C}(\mathbf{X})) > \bar{C}^* \\ 0, & \text{якщо } \text{centroid}(\tilde{C}(\mathbf{X})) > \bar{C}^*; \\ 1 - \mu_{\tilde{C}(\mathbf{X})}(\bar{C}^*), & \text{інакше} \end{cases} \quad (7.7)$$

$$\mu_{\tilde{T}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{T}(\mathbf{X})) > \bar{T}^* \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{T}(\mathbf{X})) > \bar{T}^* \\ 0, & \text{якщо } \text{centroid}(\tilde{T}(\mathbf{X})) > \bar{T}^*; \\ 1 - \mu_{\tilde{T}(\mathbf{X})}(\bar{T}^*), & \text{інакше} \end{cases} \quad (7.8)$$

$$\mu_{\tilde{I}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{I}(\mathbf{X})) > \bar{I}^* \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{I}(\mathbf{X})) > I^* \\ 0, & \text{якщо } \text{centroid}(\tilde{I}(\mathbf{X})) > I^* \\ 1 - \mu_{\tilde{I}(\mathbf{X})}(I^*), & \text{інакше} \end{cases}; \quad (7.9)$$

$$\mu_{\tilde{q}_j}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{p}^{0j}(\mathbf{X})) > \bar{q}_j \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{p}^{0j}(\mathbf{X})) > q_j \\ 0, & \text{якщо } \text{centroid}(\tilde{p}^{0j}(\mathbf{X})) > q_j \\ 1 - \mu_{\tilde{p}^{0j}(\mathbf{X})}(q_j), & \text{інакше} \end{cases}; \quad (7.10)$$

$$\mu_{\tilde{S}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } S(\mathbf{X}) > S^* \\ 1, & \text{якщо } S(\mathbf{X}) \leq S^* \end{cases}; \quad (7.11)$$

де *centroid* – дефаззіфікація за центром тяжіння;

*core* та *core* – мінімальне та максимальне значення ядра нечіткого числа.

В (7.6)–(7.10) перший випадок нульового значення ступеня належності відповідає порушенню граничного обмеження з постановки задачі оптимізації (7.5). Другий випадок нульового значення ступеня належності відповідає ситуації, коли найбільш можливе значення показника надійності не задовольняє мету оптимізації. Третій випадок нульового значення ступеня належності відповідає ситуації, коли значення показника надійності в середньому (за центром тяжіння нечіткого числа) не задовольняє ціль оптимізації. Ця умова необов'язкова – введення її в (7.6)–(7.10) реалізує песимістичний підхід щодо ризикованості при прийнятті рішень. Рис. 7.6 на прикладі формули (7.12) пояснює розрахунок ступеня задоволення чіткої мети оптимізації альтернативою  $\mathbf{X}$  з нечітким значенням показника надійності АП за песимістичного підходу щодо ризикованості при прийнятті рішення.

Як і в попередньому випадку (підрозділ 7.2) якщо в задачі оптимізації якийсь з критеріїв не застосовано, тоді відповідний ступінь належності прирівнюємо до 1. Якщо значення характеристики надійності АП представлено чітким числом, тоді ступінь виконання відповідного критерію розраховуємо за аналогією з формулою (7.11).

Для визначення ступеня досягнення мети за всіма критеріями одночасно, підставимо розраховані за (7.6)–(7.11) значення в формули (7.1) або (7.2). Вибір кращого розв'язку ( $\mathbf{X}$ ) з альтернатив  $\mathbf{X}_1$  та  $\mathbf{X}_2$  здійснимо за таким алгоритмом:

```

If  $\mu_{\tilde{G}}(\mathbf{X}_1) > \mu_{\tilde{G}}(\mathbf{X}_2)$  Then  $\mathbf{X} := \mathbf{X}_1$ ;
elseif  $\mu_{\tilde{G}}(\mathbf{X}_2) > \mu_{\tilde{G}}(\mathbf{X}_1)$  Then  $\mathbf{X} := \mathbf{X}_2$ ;
elseif  $\mu_{\tilde{G}}(\mathbf{X}_1) > 0$  Then
    <обрати альтернативу за більшим значенням ступеня
    виконання найважливішого критерію>;
elseif  $E(\mathbf{X}_1) < E(\mathbf{X}_2)$  Then  $\mathbf{X} := \mathbf{X}_1$ ;
elseif  $E(\mathbf{X}_1) > E(\mathbf{X}_2)$  Then  $\mathbf{X} := \mathbf{X}_2$ ;
else <обрати альтернативу за більшим значенням ступеня виконан-
    ня найважливішого критерію>;
end

```

Відмінність цього алгоритму від попереднього з підрозділу 7.2 полягає в порівнянні альтернатив не за абсолютним значенням найважливішого критерію, а за ступенем його виконання. Це зроблено, щоб не порівнювати нечіткі числа, якими в даному випадку задані абсолютні значення критеріїв. Якщо ці ступені належності однакові, тоді порівнюємо за результатами дефазифікації нечіткого значення найважливішого критерію. Якщо і ці числа будуть однаковими, тоді рішення приймемо за критерієм з другим рангом важливості і т.д.

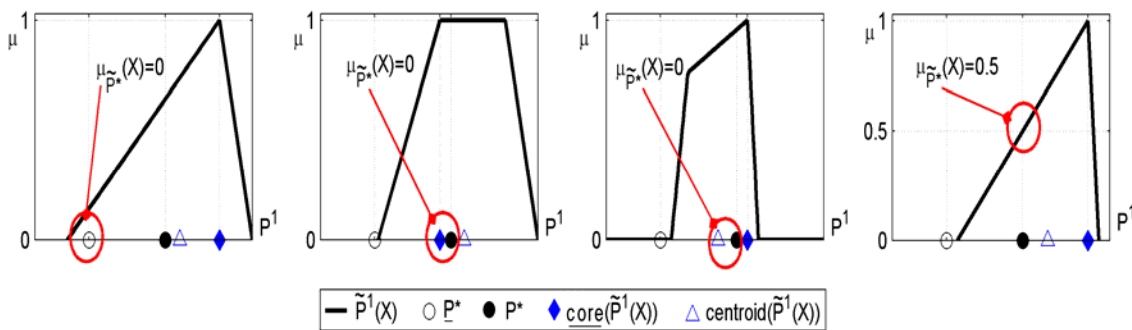


Рис. 7.5. До формули (7.6)

**Приклад 7.4.** Розглядається задача вибору варіантів реалізації операторів АП з нечіткими характеристиками надійності. АП складається з послідовного виконання двох операторів –  $Y = A_1 A_2$ , кожен з яких можна реалізувати двома способами:  $A_1 \in \{A_{1\_1}, A_{1\_2}\}$  та  $A_2 \in \{A_{2\_1}, A_{2\_2}\}$ . Нечіткі характеристики надійності реалізацій операторів АП зведені в табл. 7.2. Проранжуємо варіанти АП за такої мети оптимізації:

- $P^* = 0,94$ ;  $\overline{P}^* = 0,9$ ;
- $C^* = 14$ ;  $\overline{C}^* = 22$ ;
- $T^* = 10$ ;  $\overline{T}^* = 12$ .



Як і в прикладі 7.3 задачу розв'яжемо повним перебором можливих варіантів АП. Цим варіантам АП поставимо у відповідність такі вектори керованих змінних:

$$A_{1\_1}A_{2\_1} \rightarrow \mathbf{X}=(1, 1);$$

$$A_{1\_1}A_{2\_2} \rightarrow \mathbf{X}=(1, 2);$$

$$A_{1\_2}A_{2\_1} \rightarrow \mathbf{X}=(2, 1);$$

$$A_{1\_2}A_{2\_2} \rightarrow \mathbf{X}=(2, 2).$$

Таблиця 7.2

Параметри функцій належності характеристик надійності операторів з прикладу 7.2

$A$	$\tilde{p}^1$ (трикутна функція належності)	$\tilde{C}$ (трапецієвидна функція належності)	$\tilde{T}$ (трикутна функція належності)
$A_{1\_1}$	$a=0,93; b=0,97; c=1$	$a=4; b=5; c=5,5; d=7$	$a=1; b=2; c=7$
$A_{1\_2}$	$a=0,96; b=0,98; c=0,99$	$a=5; b=6; c=7; d=12$	$a=3; b=4; c=5$
$A_{2\_1}$	$a=0,97; b=0,98; c=1$	$a=3; b=4; c=5; d=10$	$a=2; b=5; c=7$
$A_{2\_2}$	$a=0,95; b=0,97; c=0,99$	$a=2; b=4; c=5; d=8$	$a=1; b=2; c=4$

Нечіткі показники надійності кожного варіанта АП розрахуємо за нечіткими моделями послідовної структури [82]. Для цього нечіткі числа представимо сукупністю  $\alpha$ -зрізів на  $\alpha$ -рівнях 0, 0,25, 0,5, 0,75 та 1. Розрахунки зведемо в табл. 7.3 та в табл. 7.4.

Підставляючи ступені належності з рис. 7.6 в (7.1) отримуємо:

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,217 \wedge 0,539 \wedge 0 = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0 \wedge 0,778 \wedge 0,857 = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 0,697 \wedge 0,2 \wedge 0,333 = 0,2;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 0,273 \wedge 0,25 \wedge 1 = 0,25.$$

За алгоритмом прийняття рішення отримуємо, що кращою є альтернатива  $\mathbf{X}=(2, 2)$ , а другою за рангом –  $\mathbf{X}=(2, 1)$ . Решта альтернатив порушує по одній меті оптимізації (рис. 7.6). Проранжуємо їх, вважаючи за домінуючий критерій ймовірність відсутності помилки на виході АП. Для альтернативи  $\mathbf{X}=(1, 1)$  результат дефаззифікації нечіткого числа  $\tilde{p}^1(\mathbf{X})$  більший, ніж для альтернативи  $\mathbf{X}=(1, 2)$  (див. табл. 7.5). Відповідно третій ранг надамо альтернативі  $\mathbf{X}=(1, 1)$ , а четвертий ранг альтернативі  $\mathbf{X}=(1, 2)$ .

Проранжуємо альтернативи за таких коефіцієнтів важливості критеріїв:  $w_P=0,7$ ,  $w_C=0,1$  та  $w_T=0,2$ . За (7.2) отримуємо

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,217^{0.7} \wedge 0,539^{0.1} \wedge 0^{0.2} = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0^{0.7} \wedge 0,778^{0.1} \wedge 0,857^{0.2} = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 0,697^{0.7} \wedge 0,2^{0.1} \wedge 0,333^{0.2} = 0,777;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 0,273^{0.7} \wedge 0,25^{0.1} \wedge 1^{0.2} = 0,403.$$

Відповідно альтернативи впорядковуємо такими чином:

1-ий ранг для  $\mathbf{X}=(2, 1)$ ;

2-ий ранг для  $\mathbf{X}=(2, 2)$ ;

3-ій ранг для  $\mathbf{X}=(1, 1)$ ;

4-ий ранг для  $\mathbf{X}=(1, 2)$ . ☒

Таблиця 7.3

До прикладу 7.4: нечіткі показники надійності операторів в  $\alpha$ -формі

Оператор	$\alpha$	$\underline{P}_\alpha$	$\overline{P}_\alpha$	$\underline{C}_\alpha$	$\overline{C}_\alpha$	$\underline{T}_\alpha$	$\overline{T}_\alpha$
$A_{1\_1}$	1	0,97	0,97	5	5,5	2	2
	0,75	0,96	0,9775	4,75	5,875	1,75	3,25
	0,5	0,95	0,985	4,5	6,25	1,5	4,5
	0,25	0,94	0,9925	4,25	6,625	1,25	5,75
	0	0,93	1	4	7	1	7
$A_{1\_2}$	1	0,98	0,98	6	7	4	4
	0,75	0,975	0,9825	5,75	8,25	3,75	4,25
	0,5	0,97	0,985	5,5	9,5	3,5	4,5
	0,25	0,965	0,9875	5,25	10,75	3,25	4,75
	0	0,96	0,99	5	12	3	5
$A_{2\_1}$	1	0,98	0,98	4	5	5	5
	0,75	0,9775	0,985	3,75	6,25	4,25	5,5
	0,5	0,975	0,99	3,5	7,5	3,5	6
	0,25	0,9725	0,995	3,25	8,75	2,75	6,5
	0	0,97	1	3	10	2	7
$A_{2\_2}$	1	0,97	0,97	4	5	2	2
	0,75	0,965	0,975	3,5	5,75	1,75	2,5
	0,5	0,96	0,98	3	6,5	1,5	3
	0,25	0,955	0,985	2,5	7,25	1,25	3,5
	0	0,95	0,99	2	8	1	4

Таблиця 7.4

До прикладу 7.4: нечіткі характеристики надійності варіантів АП

$\mathbf{X}$	$\alpha$	$\underline{P}_\alpha$	$\overline{P}_\alpha$	$\underline{C}_\alpha$	$\overline{C}_\alpha$	$\underline{T}_\alpha$	$\overline{T}_\alpha$
(1, 1)	1	0,9506	0,9506	9	10,5	7	7
	0,75	0,9384	0,96284	8,5	12,125	6	8,75
	0,5	0,92625	0,97515	8	13,75	5	10,5
	0,25	0,91415	0,98754	7,5	15,375	4	12,25
	0	0,9021	1	7	17	3	14
(1, 2)	1	0,9409	0,9409	9	10,5	4	4
	0,75	0,9264	0,95306	8,25	11,625	3,5	5,75
	0,5	0,912	0,9653	7,5	12,75	3	7,5
	0,25	0,8977	0,97761	6,75	13,875	2,5	9,25
	0	0,8835	0,99	6	15	2	11
(2, 1)	1	0,9604	0,9604	10	12	9	9
	0,75	0,95306	0,96776	9,5	14,5	8	9,75
	0,5	0,94575	0,97515	9	17	7	10,5
	0,25	0,93846	0,98256	8,5	19,5	6	11,25
	0	0,9312	0,99	8	22	5	12
(2, 2)	1	0,9506	0,9506	10	12	6	6
	0,75	0,94087	0,95794	9,25	14	5,5	6,75
	0,5	0,9312	0,9653	8,5	16	5	7,5
	0,25	0,92157	0,97269	7,75	18	4,5	8,25
	0	0,912	0,9801	7	20	4	9

Таблиця 7.5

До прикладу 7.4: результати дефаззифікації нечітких показників надійності варіантів АП

$\mathbf{X}$	$centroid(\tilde{p}^1(\mathbf{X}))$	$centroid(\tilde{C}(\mathbf{X}))$	$centroid(\tilde{T}(\mathbf{X}))$
(1, 1)	0,95066	10,313	7,4688
(1, 2)	0,93949	9,9375	4,7813
(2, 1)	0,96043	12	8,8438
(2, 2)	0,94913	11,625	6,1563

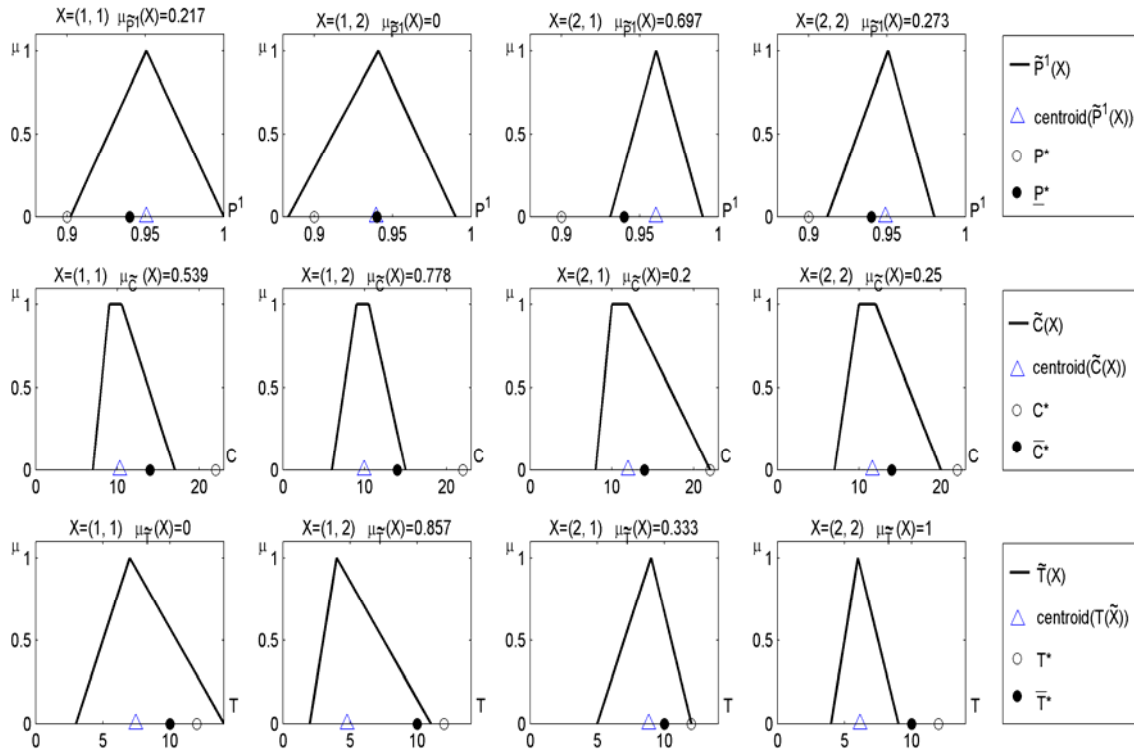


Рис. 7.6. Виконання нечіткими показниками надійності варіантів АП мети оптимізації в прикладі 7.4

#### 7.4. Оптимізації за нечіткої мети та нечітких початкових даних

Розглянемо випадок, коли і значення характеристик надійності елементів АП, і мета оптимізації задані нечіткими числами. Задачу оптимізації надійності АП в цьому випадку поставимо таким чином: знайти вектор  $\mathbf{X}$ , щоб

$$\left\{ \begin{array}{l}
 \tilde{p}^1(\mathbf{X}) = \tilde{P}^*; \\
 \tilde{p}^{0j}(\mathbf{X}) = \tilde{q}_j, \quad j = \overline{1, m}; \\
 \tilde{C}(\mathbf{X}) = \tilde{C}^*; \\
 \tilde{T}(\mathbf{X}) = \tilde{T}^*; \\
 \tilde{I}(\mathbf{X}) = \tilde{I}^*; \\
 S(\mathbf{X}) = \tilde{S}^*; \\
 \underline{\text{supp}}(\tilde{p}^1(\mathbf{X})) \geq \underline{\text{supp}}(\tilde{P}^*); \\
 \overline{\text{supp}}(\tilde{p}^{0j}(\mathbf{X})) \leq \overline{\text{supp}}(\tilde{p}^{0j}(\mathbf{X})), \quad j = \overline{1, m}; \\
 \overline{\text{supp}}(\tilde{C}(\mathbf{X})) \leq \overline{\text{supp}}(\tilde{C}(\mathbf{X})); \\
 \overline{\text{supp}}(\tilde{T}(\mathbf{X})) \leq \overline{\text{supp}}(\tilde{T}^*); \\
 \overline{\text{supp}}(\tilde{I}(\mathbf{X})) \leq \overline{\text{supp}}(\tilde{I}^*)
 \end{array} \right. \quad (7.12)$$

за умови, що ядра цільових нечітких чисел  $\tilde{P}^*$ ,  $\tilde{C}^*$ ,  $\tilde{T}^*$ ,  $\tilde{q}_j$  ( $j = \overline{1, m}$ ),  $\tilde{I}^*$  та  $\tilde{S}^*$  задані з (7.3).

Як і в (7.4), так і в (7.12) чіткі обмеження–нерівності визначають чітку область допустимих розв’язків. Порухення будь-якого з них робить розв’язок недопустимим.

Ступені досягнення варіантом  $X$  мети оптимізації за кожним критерієм розрахуємо таким чином:

$$\mu_{\tilde{P}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{p}^1(\mathbf{X})) < \overline{\text{supp}}(\tilde{P}^*) \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{p}^1(\mathbf{X})) < P^* \\ 0, & \text{якщо } \text{centroid}(\tilde{p}^1(\mathbf{X})) < P^* \\ 1 - \max_{u \leq P^*}(\Delta\mu_P(u)), & \text{інакше} \end{cases} \quad (7.13)$$

$$\mu_{\tilde{C}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{C}(\mathbf{X})) > \overline{\text{supp}}(\tilde{C}^*) \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{C}(\mathbf{X})) > C^* \\ 0, & \text{якщо } \text{centroid}(\tilde{C}(\mathbf{X})) > C^* \\ 1 - \max_{u \geq C^*}(\Delta\mu_C(u)), & \text{інакше} \end{cases} ; \quad (7.14)$$

$$\mu_{\tilde{T}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{T}(\mathbf{X})) > \overline{\text{supp}}(\tilde{T}^*) \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{T}(\mathbf{X})) > T^* \\ 0, & \text{якщо } \text{centroid}(\tilde{T}(\mathbf{X})) > T^* \\ 1 - \max_{u \geq T^*}(\Delta\mu_T(u)), & \text{інакше} \end{cases} ; \quad (7.15)$$

$$\mu_{\tilde{I}^*}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{I}(\mathbf{X})) > \overline{\text{supp}}(\tilde{I}^*) \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{I}(\mathbf{X})) > I^* \\ 0, & \text{якщо } \text{centroid}(\tilde{I}(\mathbf{X})) > I^* \\ 1 - \max_{u \geq I^*}(\Delta\mu_I(u)), & \text{інакше} \end{cases} ; \quad (7.16)$$

$$\mu_{q_j}(\mathbf{X}) = \begin{cases} 0, & \text{якщо } \overline{\text{supp}}(\tilde{p}^{0j}(\mathbf{X})) > \overline{\text{supp}}(\tilde{q}_j) \\ 0, & \text{якщо } \overline{\text{core}}(\tilde{p}^{0j}(\mathbf{X})) > q_j \\ 0, & \text{якщо } \text{centroid}(\tilde{p}^{0j}(\mathbf{X})) > q_j \\ 1 - \max_{u \geq q_j}(\Delta\mu_{\tilde{p}^{0j}}(u)), & \text{інакше} \end{cases} \quad (7.17)$$

$$\mu_{\tilde{S}^*}(\mathbf{X}) = \mu_{\tilde{S}^*}(S(\mathbf{X})), \quad (7.18)$$

де

$$\Delta\mu_C(u) = \mu_{\tilde{C}(\mathbf{X})}(u) - \mu_{\tilde{C}^*}(u);$$

$$\Delta\mu_T(u) = \mu_{\tilde{T}(\mathbf{X})}(u) - \mu_{\tilde{T}^*}(u);$$

$$\Delta\mu_I(u) = \mu_{\tilde{I}(\mathbf{X})}(u) - \mu_{\tilde{I}^*}(u);$$

$$\Delta\mu_{p^0j}(u) = \mu_{p^0j(\mathbf{X})}(u) - \mu_{qj}(u);$$

$$\Delta\mu_p(u) = \mu_{\tilde{p}^1(\mathbf{X})}(u) - \mu_{\tilde{p}^*Pu}.$$

В (7.13)–(7.17) перший випадок нульового значення ступеня належності відповідає порушенню граничного обмеження з постановки задачі оптимізації (7.12). Другий випадок нульового значення ступеня належності відповідає ситуації, коли найбільш можливе значення показника надійності АП гірше найбільш можливого значення цільового нечіткого числа. Третій випадок нульового значення ступеня належності відповідає ситуації, коли значення показника надійності АП в середньому (за центром тяжіння нечіткого числа) гірше найбільш можливого значення цільового нечіткого числа. Ця умова є обов’язковою – введення її в (7.13)–(7.17) реалізує песимістичний підхід щодо ризикованості при прийнятті рішень. Рис. 7.7 на прикладі формули (7.13) пояснює розрахунок ступеня задоволення нечіткої мети оптимізації альтернативою  $\mathbf{X}$  з різними нечіткими значеннями показника надійності АП при песимістичному підході щодо ризикованості при прийнятті рішення. На рис. 7.7г ступінь належності розраховано таким чином:  $\mu_{\tilde{p}^*}(\mathbf{X}) = 1 - (0,75 - 0,43) = 0,68$ .

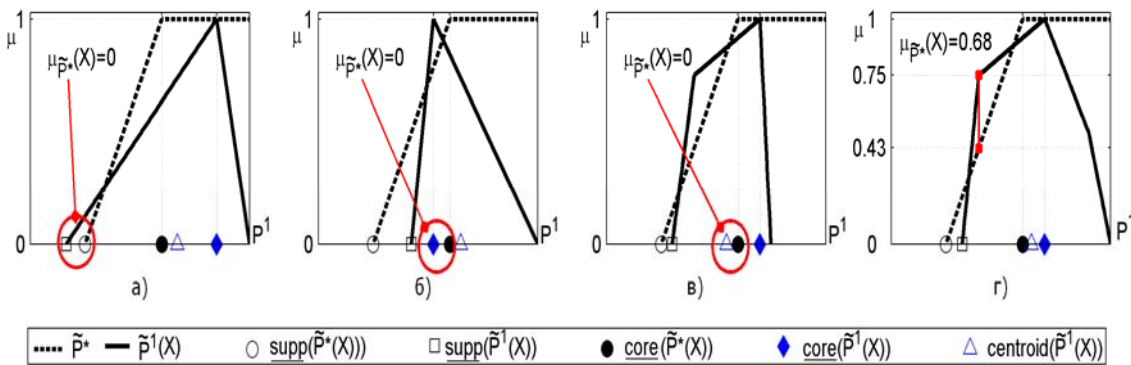


Рис. 7.7. До формули (7.13)

Після розрахунків за (7.13)–(7.18) результати підставляємо в формулу (7.7) або (7.8) та приймаємо рішення так само, як і в попередньому випадку, тобто за алгоритмом з підрозділу 7.5.

*Приклад 7.5.* Проранжуємо варіанти АП з прикладу 7.3, якщо їх показники надійності описані нечіткими числами з табл. 7.4.

Порівняння нечітких показників надійності варіантів АП з цільовими нечіткими числами наведено на рис. 7.8. При розрахунку ступеня досягнення нечіткої мети оптимізації за (7.13)–(7.15) застосовано лінійну інтерполяцію функції належності за вузлами  $\alpha$ -зрізів нечітких чисел  $\tilde{p}^1(\mathbf{X})$ ,  $\tilde{C}(\mathbf{X})$  та  $\tilde{T}(\mathbf{X})$ . Ступені належності розраховані за оптимістичним підходом врахування ризику при прийнятті рішення, тобто формули (7.13)–(7.15) застосовані без обмеження на значення результату дефазифікації нечітких показників надійності АП.

Підставляючи ступені належності з рис. 7.9 в (7.7), отримуємо

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,723 \wedge 1 \wedge 0 = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0 \wedge 1 \wedge 1 = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 1 \wedge 0,734 \wedge 0,669 = 0,669;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 0,864 \wedge 0,873 \wedge 1 = 0,864.$$

За алгоритмом з підрозділу 7.3 знаходимо, що кращою є альтернатива  $\mathbf{X}=(2, 2)$ , яка відповідає АП  $Y = A_{1\_2}A_{2\_2}$ . Другою за рангом є альтернатива  $\mathbf{X}=(2, 1)$ . Решта альтернатив порушує по одній нечіткій меті (рис. 7.8), тому для їх ранжування потрібно знати домінуючий критерій. Нехай, як і в попередніх прикладах, ним буде ймовірність відсутності помилки на виході АП. Для альтернативи  $\mathbf{X}=(1, 1)$  результат дефазифікації нечіткого числа  $\tilde{p}^1(\mathbf{X})$  дорівнює 0,951, а для альтернативи  $\mathbf{X}=(1, 2)$  – 0,938. Відповідно, третій ранг надаємо альтернативі  $\mathbf{X}=(1, 1)$ , а четвертий ранг – альтернативі  $\mathbf{X}=(1, 2)$ .

Проранжуємо альтернативи за такими коефіцієнтами важливості критеріїв:  $w_p=0,7$ ,  $w_c=0,1$  та  $w_t=0,2$ . За (7.2) отримуємо

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 1)) = 0,723^{0.7} \wedge 1^{0.1} \wedge 0^{0.2} = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (1, 2)) = 0^{0.7} \wedge 1^{0.1} \wedge 1^{0.2} = 0;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 1)) = 1^{0.7} \wedge 0,734^{0.1} \wedge 0,669^{0.2} = 0,923;$$

$$\mu_{\tilde{G}}(\mathbf{X} = (2, 2)) = 0,864^{0.7} \wedge 0,873^{0.1} \wedge 1^{0.2} = 0,903.$$

В цьому випадку альтернативи впорядковуємо такими чином: 1-ий ранг для  $\mathbf{X}=(2, 1)$ ; 2-ий ранг для  $\mathbf{X}=(2, 2)$ ; 3-ій ранг для  $\mathbf{X}=(1, 1)$ ; 4-ий ранг для  $\mathbf{X}=(1, 2)$ .  $\square$

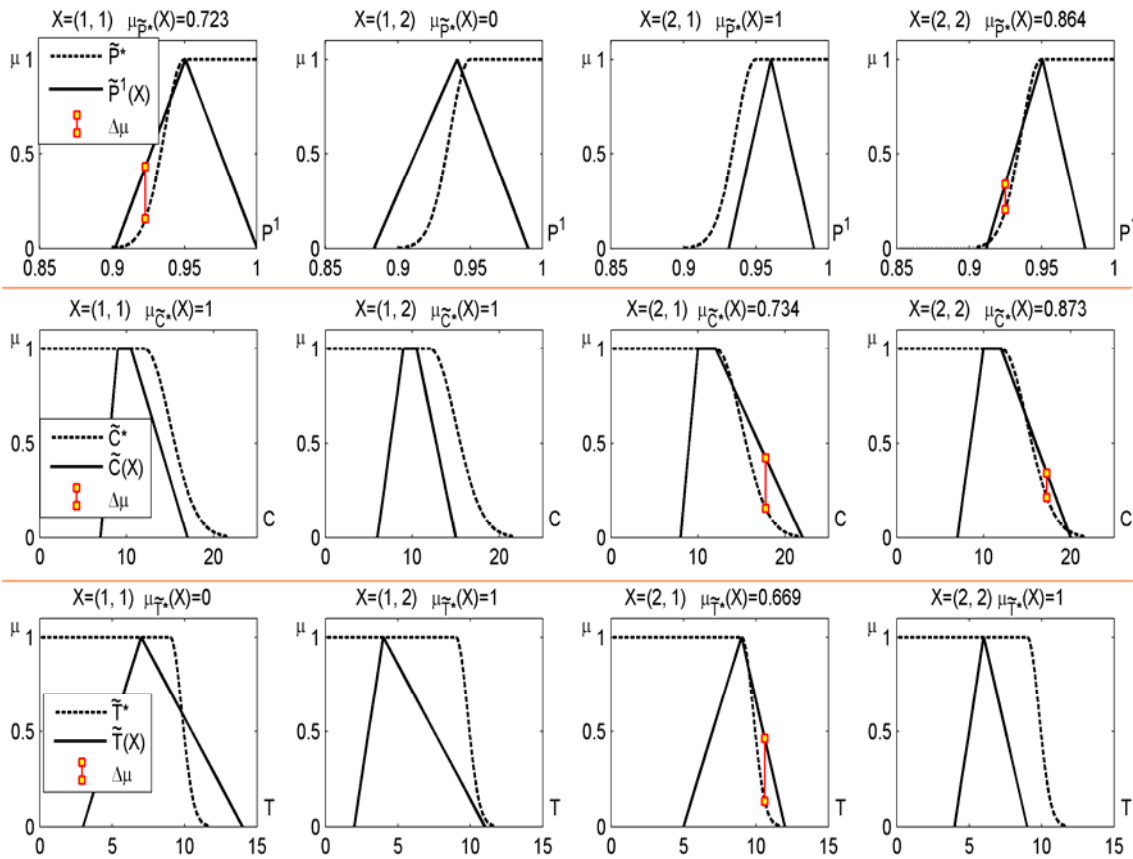


Рис. 7.8. Виконання розв'язками нечіткої мети оптимізації в прикладі 7.5

### 7.5. Особливості генетичного алгоритму нечіткої оптимізації

Нечітку оптимізацію надійності АП за постановками (7.4), (7.5) та (7.12) за будь-яких керованих змінних може здійснити ГА з розділу 5. Нечітка оптимізація вносить такі особливості в реалізацію генетичних операцій:

- фітнес-функцію реалізуємо за формулою (7.1) або (7.2);
  - селекцію популяції проведемо за новою турнірною схемою, порівнюючи хромосоми за алгоритмами вибору кращого розв'язку з підрозділів 7.2 та 7.3;
  - при генеруванні якісної початкової популяції чітким значенням в формулах з табл. 5.1 та табл. 5.2 поставимо у відповідність результати дефаззифікації за центром тяжіння цільових нечітких чисел та нечітких початкових даних про характеристики надійності операторів та логічних умов АП;
- } (7.19)



- у випадку нечітких характеристик надійності операторів та логічних умов розрахунки показників надійності варіанта АП проведемо за моделями типових структур з розділу 6. } (7.20)

Генетичне кодування варіанта АП, операції мутації та схрещення залишимо без будь-яких змін. Нижче наведено ГА нечіткої оптимізації надійності АП для загального випадку за нечіткої мети та нечітких початкових даних.

*⟨Ініціалізувати параметри генетичного алгоритму:*

*pop\_size – розмір популяції; num\_iter – кількість ітерацій;*

*num\_cross – кількість схрещувань за одну ітерацію;*

*num\_mut – кількість мутацій за одну ітерацію⟩*

*⟨Згенерувати якісну початкову популяцію за (7.19)⟩*

*⟨Розрахувати показники надійності закодованих варіантів за (7.20)⟩*

*⟨Розрахувати фітнес-функцію нових хромосом за схемою Беллмана-Заде згідно з формулами (7.1) або (7.2)⟩*

**For**  $t=1:num\_iter$

*⟨Виконати num\_cross схрещень хромосом⟩*

*⟨Виконати num\_mut мутацій хромосом⟩*

*⟨Для нових хромосом розрахувати показники надійності за (7.20)⟩*

*⟨Розрахувати фітнес-функцію нових хромосом за схемою Беллмана–Заде згідно з формулою (7.1) або (7.2)⟩*

*⟨Зафіксувати елітну хромосому з найкращою фітнес-функцією⟩*

*⟨Включити елітну хромосому в нову популяцію⟩*

*⟨За алгоритмами з підрозділу 5.6 провести pop\_size-1 турнірів⟩*

*⟨Сформувати нову популяцію з переможців турнірів⟩*

**End**

*⟨Декодувати елітну хромосому⟩.*

## ПІСЛЯМОВА

Багатовимірні АП дозволяють формалізувати дискретні процеси у вигляді алгоритму функціонування, що описує події внесення, виявлення та усунення помилок різних типів. Помилки різних типів виникають при неправильному виконанні людиною або технікою деякого фрагменту алгоритму. До таких процесів можна віднести технологічні процеси в радіоелектронній, приладобудівній, харчовій промисловостях та алгоритми діяльності операторів АЕС, повітряно-космічних систем тощо.

В книзі поставлено задачу моделювання та оптимізації надійності багатовимірних АП та розроблено комплекс моделей та алгоритмів для їхнього розв'язання.

Моделювання надійності здійснено за середнім часом, середньою вартістю та за ймовірністю безпомилкового виконання багатовимірного АП, а також за ймовірностями виникнення на виході таких процесів помилок кожного типу. Розроблені моделі надійності дозволяють оцінювати ці показники для чітких і нечітких початкових даних.

Для забезпечення надійності багатовимірних АП розроблено градієнтні та генетичні алгоритми оптимізації, які дозволяють достатньо швидко знаходити оптимальну структуру АП. Розроблений комплекс алгоритмів забезпечує необхідні рівні основних показників надійності як для чітких так і для нечітких початкових даних.

Автори сподіваються, що читач знайшов в цій книзі відповіді на такі запитання:

1. Що таке багатовимірний АП ?
2. Як оцінювати надійність багатовимірних АП для чітких і нечітких початкових даних ?
3. Як застосовувати моделі надійності багатовимірних АП ?
4. Як забезпечити необхідні рівні надійності багатовимірних АП ?
5. Як проводити оптимізацію надійності АП для нечітких початкових даних ?
6. Як прискорити алгоритми оптимізації надійності ?

## Література

1. Ашеров А.Т. Сажко Г.І. Ергономіка інформаційних технологій: оцінка, проектування, експертиза: Навч. посіб. – Харків: УПА, 2005. – 243 с.
2. Байцер Б. Микроанализ производительности вычислительных систем: Пер. с англ. – М.: Радио и связь, 1983. –360 с.
3. Банковские системы, бюджетные и финансовые технологии компании Банк'с Софт Системс: [www.bssys.com](http://www.bssys.com).
4. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. // Вопросы анализа и процедуры принятия решений. – М.: Мир., 1976. – С.172–215.
5. Бібліотека тестових задач [www.ksu.vstu.vinnica.ua/shtovba/benchmarks](http://www.ksu.vstu.vinnica.ua/shtovba/benchmarks)
6. Бодров В.А., Орлов В.Я. Психология и надежность: человек в системах управления техникой. – М.: РАН. Ин-т психологии. – 1998. – 285 с.
7. Бондарь Ю.В., Сафонов И.В. Об одном методе оптимального использования алгоритмической избыточности // Автоматика и вычислительная техника. – 1975. – №3. – С.26–29.
8. Вентцель Е.С. Исследование операций: Задачи, принципы, методология. –М.: Наука, 1988. – 75с.
9. Вентцель Е.С. Исследование операций. М.: Сов. радио, 1972. – 554 с.
10. Вигман Б.А. Исследование стохастических моделей контроля качества изделий (полуфабрикатов) в поточном производстве //«Стабильность и надежность информационных устройств и систем» – К.: Техніка, 1975. – С.33–38.
11. Вигман Б.А. Стохастические модели контроля // Управляющие системы и машины.– 1973. – №2. – С.112–115.
12. Вострецов А.Г., Кушнир В.И. Расстановка контроля в технологических процессах изготовления РЭА // Изв. Вузов. Сер. Радиоэлектроника. –1987. – Т. 30, №9. – С.39-42.
13. Вошинин А.П., Сотиров Г.Р. Оптимизация в условиях неопределенности. – Изд-во МЭП (СССР), «Техника» (НРБ), 1989. –224с.
14. Гвоздик М.И., Евграфов В.Г., Цой Е.Б. Оптимизация организационно-технических систем ВМФ. Методы. Алгоритмы. Программы. – Л.: ВВМУРЭ, 1997. – 223 с.
15. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. – Харьков: Основа, 1997.–212 с.

16. Герасимов Б.М., Тарасов В.А., Токарев И.В. Человеко-машинные системы принятия решений с элементами искусственного интеллекта. – К.: Наукова думка, 1993. – 180 с.
17. Герасимов Б.М., Дивизнюк М.М., Субач И.Ю. Системы поддержки принятия решений: проектирование, применение, оценка эффективности. – Севастополь: Научно-исследовательский центр вооруженных сил Украины «Государственный океанариум», 2004. – 320 с.
18. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – К.: Наук. Думка, 1978. – 320 с.
19. Гриф М.Г., Цой Е.Б. Методы последовательной оптимизации управляющих человеко-машинных комплексов на основе вероятностного и нечеткого моделирования // Научный вестник Новосибирского национального технического университета. – 1999. – №5. – С. 21–41.
20. Гриф М.Г., Цой Е.Б. Автоматизация проектирования процессов функционирования человеко-машинных систем на основе метода последовательной оптимизации. Новосибирск: Изд-во НГТУ, 2005. – 264 с.
21. Гриф М.Г., Цой Е.Б. Последовательная оптимизация эрготехнических систем на основе аппарата функциональных сетей. – К.: Знание, 1989. – 16 с.
22. Губинский А.И. Надежность и качество функционирования эргатических систем. – Л.: Наука, 1982. – 270 с.
23. Губинский А.И., Евграфов В.Г. Эргономическое проектирование судовых систем управления. – Л.: Судостроение, 1977. – 224 с.
24. Губинский А.И., Гречко Ю.П., Ротштейн А.П. Методические рекомендации по построению аналитических моделей оценки надежности и качества функционирования эргатических систем. Ч. II. – М.: Из-во Научного совета по комплексной проблеме «Кибернетика» АН СССР, 1978. – 77 с.
25. Губинский А.И., Ротштейн А.П. Методические рекомендации по построению моделей оценки эффективности, качества и надежности эрготехнических систем. – М.: Из-во Научного совета по комплексной проблеме «Кибернетика» АН СССР, 1981. – 66 с.
26. Губинский А.И., Ротштейн А.П. Применение обобщенного структурного метода для оценки диалоговых систем «человек–ЭВМ» // Диалоговые системы. – 1980. – Вып. 3 «Реализация диалоговых систем». – Рига: Зинатне. – С. 31–49.
27. Данциг Дж. Линейное программирование, его обобщение и применения: Пер. с англ.–М.: Прогресс, 1965.– 600 с.

28. Дехтярев Ю.И. Методы оптимизации. – М.: Сов. радио, 1980. – 270 с.
29. Дружинин Г.В. Анализ эрготехнических систем. – М.: Энергоатомиздат, 1984. – 160 с.
30. Дружинин Г.В. Методы оценки и прогнозирования качества. – М.: Радио и связь, 1982. – 160 с.
31. Дружинин Г.В. Надежность автоматизированных производственных систем. – М.: Энергоатомиздат, 1986. – 480 с.
32. Душков Б.А., Королев А.В., Смирнов Б.А. Основы инженерной психологии: Учебник для вузов. – М.: Академический Проект; Екатеринбург: Деловая книга, 2002. – 576 с.
33. Дюк В., Самойленко А. Data mining: учебный курс. – СПб: Питер, 2001. – 368 с.
34. Жилинскас А., Шалтянис В. Поиск оптимума. – М.: Наука, 1989. – 48 с.
35. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976. – 165 с.
36. Зайченко Ю.П. Исследование операций: Учебник. – 6-е изд., перераб. и доп. – К.: Изд. дом «Слово», 2003. – 688 с.
37. Зайченко Ю.П. Многокритериальные задачи принятия решений в нечетких условиях и методы их решения // Системні дослідження та інформаційні технології. – 2002. – №2. – С.53–62.
38. Заносин С.А., Аршакуни В.Л. Оптимизация размещения контрольных точек в технологическом процессе механической обработки // Надежность и контроль качества. – 1988. – №6. – С.42–46.
39. Зараковский Г., Медведев В. Классификация ошибок оператора // Техническая эстетика. – 1971. – №10. – С.11–17.
40. Зараковский Г.М., Павлов В.В. Закономерности функционирования эрготехнических систем. – М.: Радио и связь, 1987. – 232 с.
41. Згуровский М.З. Интегрированные системы оптимального управления и проектирования: Учеб. пособие. – К.: Вища школа, 1990. – 351 с.
42. Зингер Н.С., Куцык Б.С. Обеспечение достоверности данных в автоматизированных системах управления производством. – М.: Наука, 1974. – 136 с.
43. Информационно-управляющие человеко-машинные системы: Исследование, Проектирование, Испытания: Справочник / Адаменко А.Н., Губинский А.И., Ротштейн А.П. и др. – М.: Машиностроение, 1993. – 528 с.

44. Козлов Б.А., Ушаков И.А. Справочник по расчету надежности аппаратуры радиоэлектроники и автоматики. – М.: Сов. радио, 1975. – 472 с.
45. Компьютер и задачи выбора. – М.: Наука, 1989. – 110 с.
46. Коршунов Ю.М. Математические основы кибернетики. – М.: Энергия, 1980. – 421 с.
47. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1975. – 472 с.
48. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988. – 480 с.
49. Кузьмин И.В. Оценка эффективности и оптимизация АСКУ. – М.: Сов. радио, 1971. – 294 с.
50. Кузьмин И.В., Березюк Н.Т., Фурманов К.К., Шаронов В.Б. Синтез вычислительных алгоритмов управления и контроля. К.: Техніка, 1975. – 248 с.
51. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб: БХВ-Петербург, 2005. – 736 с.
52. Лопухин В.А., Шафранский В.С. Автоматизация и оптимизация контроля в производстве радиодеталей. – Л.: Энергия, 1980. – 160 с.
53. Лю Б. Теория и практика неопределенного программирования: Пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2005. – 416 с.
54. Мамиконов А.Г., Кульба В.В., Шелков А.Б. Достоверность, защита и резервирование информации. – М.: Энергоатомиздат, 1986. – 304 с.
55. Митюшкин Ю.И., Мокин Б.И., Ротштейн А.П. Soft-Computing: идентификация закономерностей нечеткими базами знаний. – Винница: УНІВЕРСУМ-Вінниця, 2002. – 145 с.
56. Мунипов В.М., Зинченко В.П. Эргономика. – М.: Логос, 2001. – 356 с.
57. Надежность технических систем. Справочник / Беляев Ю.К., Богатырев В.А., Болотин В.В. и др. Под ред. Ушакова И.А. – М.: Радио и связь, 1985. – 608 с.
58. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. – М.: Наука, 1986. – 312 с.
59. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. – М.: Радио и связь, 1981. – 286 с.
60. Основы инженерной психологии. Учеб. для техн. вузов / Б.А. Душков, Б.Ф. Ломов, В.Ф. Рубахин. / Под ред. Б.Ф. Ломова. – М.: Высш. шк., 1986. – 448 с.
61. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность: Пер. с англ. – М.: Мир, 1985. – 512 с.

62. Пивоваров А.Н. Методы обеспечения достоверности информации в АСУ.– М.: Радио и связь, 1982. – 144 с.
63. Поляк Б.Т. Введение в оптимизацию. – М.: Наука, 1983. –384 с.
64. Попович П.Р., Губинский А.И., Колесников Г.М. Эргономическое обеспечение деятельности космонавтов. – М.: Машиностроение, 1985. – 272 с.
65. Поспелов Д.А. Логико-лингвистические модели в системах управления.– М.: Знание, 1985.– 48 с.
66. Прикладные нечеткие системы: Пер. с япон. / Т. Тэрано, К. Асаи, М. Сугэно и др. – М.: Мир, 1993. – 368 с.
67. Проектная оценка качества выполнения функций АСУ ГПС с учетом действий операторов АРМ: Методические рекомендации / НИИАП. – М.: ВНИИТЭМР, 1989. – 120с.
68. Растригин Л.А. Адаптация сложных систем. Методы и приложения. – Рига: Зинатне, 1981. – 375 с.
69. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике. Кн.1.: Пер. с англ. – М.: Мир, 1986. – 320 с.
70. Ротштейн А.П. Алгебраическое проектирование бездефектных трудовых процессов // Управляющие системы и машины. – 1990. – №6. – С.92–102.
71. Ротштейн А.П. Вероятностно-алгоритмические модели человеко-машинных систем // Автоматика. – 1987. – №5. – С.81–86.
72. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. – Винница: УНІВЕРСУМ–Вінниця, 1999. – 320 с.
73. Ротштейн А.П. Разработка метода количественной оценки надежности алгоритмических процессов. Автореф. дис. канд. техн. наук. Рига: Рижский политехнический институт, 1979. – 16 с.
74. Ротштейн А.П. Медицинская диагностика на нечеткой логике. – Винница: Континент–ПРИМ, 1996. – 132 с.
75. Ротштейн А.П. Об одной задаче надежностной оптимизации человеко-машинных систем // Надежность и контроль качества. – 1987. – №3. –С.39–46.
76. Ротштейн А.П. Оптимизация кратности контроля в процессах функционирования человеко-машинных систем // Надежность и контроль качества. – 1988. – №3. – С.54–60.
77. Ротштейн А.П. Проектирование процессов функционирования человеко-машинных систем. – К.: Знание, 1987. – 19 с.
78. Ротштейн А.П., Кательников Д.И. Идентификация нелинейных зависимостей нечеткими базами знаний // Кибернетика и системный анализ. – 1998. – №5. – С.53–61.

79. Ротштейн А.П., Кузнецов П.Д. Проектирование бездефектных человеко-машинных технологий. – К.: Техніка, 1992. – 180 с.
80. Ротштейн А.П., Штовба С.Д. Прогнозирование надежности алгоритмических процессов при нечетких исходных данных // Кибернетика и системный анализ. – 1998. – №4. – С.85–93.
81. Ротштейн О.П., Штовба С.Д. Оптимізація багатовимірних технологічних процесів генетичними алгоритмами // Вимірювальна та обчислювальна техніка в технологічних процесах. – 1999. – №2. – С.7–13.
82. Ротштейн А.П., Штовба С.Д. Нечеткая надежность алгоритмических процессов. – Винница: Континент – ПРИМ, 1997. – 142 с.
83. Ротштейн А.П., Штовба С.Д. Применение нечеткой базы знаний Сугено для моделирование надежности человека-оператора // Надежность. – 2007. – №1. – С. 13–20.
84. Ротштейн А.П., Штовба С.Д. Прогнозирование надежности алгоритмических процессов при нечетких исходных данных // Кибернетика и системный анализ. – 1998. – №4. – С.85-93.
85. Ротштейн О.П., Штовба С.Д. Оцінка надійності алгоритмічних процесів в умовах невизначених початкових даних // Вісник Вінницького політехнічного інституту. – 1996. – №2. – С.7-12.
86. Ротштейн О.П., Штовба С.Д., Дубіненко С.Б., Козачко О.М. Евристична оптимізація розстановки контрольних точок в технологічних процесах при багатовимірному просторі типів дефектів // Вісник Вінницького політехнічного інституту. – 2004. – №1. – С. 54–62.
87. Ротштейн А.П., Штовба С.Д., Козачко А.Н. Нечеткие модели надежности алгоритмов, учитывающие ошибки различных типов // Вестник Севастопольского государственного технического университета. Серия: Автоматизация процессов и управление. – 2004. – №58. – С.175–187.
88. Ротштейн О.П., Штовба С.Д., Козачко О.М. Градієнтна оптимізація кратностей операцій контролю технологічного процесу при обмежених ресурсах з урахуванням дефектів багатьох типів // Вісник Вінницького політехнічного інституту. – 2005. – №2. – С. 10–15.
89. Ротштейн О.П., Штовба С.Д., Козачко О.М. Нечітке прогнозування надійності алгоритмів, що враховують помилки різних типів // Вісник Вінницького політехнічного інституту. – 2005. – №4. – С.77–85.
90. Ротштейн О.П., Штовба С.Д., Козачко О.М. Оптимизация расстановки контрольных точек в многомерном технологическом про-



- цессе на базе генетических алгоритмов // Праці Міжнародної конференції з індуктивного моделювання, Львів, 2002. – Том 2. – С. 4–7.
91. Ротштейн О.П., Штовба С.Д., Козачко С.Д. Вплив якості початкової популяції на швидкість генетичної оптимізації надійності технологічних процесів // Інформаційні технології та комп'ютерна інженерія. – 2005. – №3. – С. 108–112.
  92. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. – М.: Горячая линия – Телеком. – 2006. – 452 с.
  93. Рыбаков И.Н. Оптимизация комплекса оборудования для изготовления и контроля продукции // Надежность и контроль качества. – 1984. – №5. – С. 44–50.
  94. Сафонов И.В. Надежное проектирование структурно-алгоритмических систем. – К.: КДНТП, 1975. – 37 с.
  95. Сафонов И.В. О формализованном надежном анализе алгоритмических процессов // Управляющие системы и машины. – 1973. – №6. – С.92–95.
  96. Сафонов И.В. Об алгоритмическом этапе формального надежного проектирования ЦВМ // Управляющие системы и машины. – 1972. – №1. – С.94–103.
  97. Сафонов И.В., Карась В.М. Обеспечение надежности функционирования и надежности АСУ (надежностная оптимизация алгоритмов). – К.: КДНТП, 1974. – 41 с.
  98. Суходольский Г.В. Структурно-алгоритмический анализ и синтез деятельности. – Л.: ЛГУ, 1976. – 150 с.
  99. Уткин И.Б., Шубинский И.Б. Нетрадиционные методы оценки надежности информационных систем. СПб.: Любавич, 2000. – 173 с.
  100. Ушаков И.А. Методы решения простейших задач надежности при наличии ограничений. – М.: Сов. радио, 1969. – 162 с.
  101. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей: Пер. с англ. – М.: Мир, 1984. – 496 с.
  102. Цибулевский И.Е. Ошибочные реакции человека-оператора. – М.: Сов. радио, 1979. – 208 с.
  103. Цыпкин Я.З. Основы информационной теории идентификации. – М.: Наука, 1984. – 320 с.
  104. Чабаненко П.П. Математические модели быстрогодействия «человек–техника». – К.: Знание, 1980. – 23 с.
  105. Человеческий фактор. Т. 5. Эргономические основы проектирования рабочих мест: Пер. с англ. / К. Кремер, Д. Чэффин, М. Айюб и др. – М.: Мир, 1992. – 390 с.

106. Шеридан Т.Б., Феррел У.Р. Системы человек-машина: модели обработки информации и принятия решений человеком-оператором. – М.: Машиностроение, 1980. – 400 с.
107. Шибанов Г.П. Количественные характеристики человека-оператора. М.: Машиностроение, 1983. – 230 с.
108. Штовба С.Д. Муравьиные алгоритмы: теория и применение // Программирование. – №4. – 2005. – С. 1–16.
109. Штовба С.Д., Козачко О.М. Генетична оптимізація кратностей контрольно-добробчих операцій в технологічних процесах з урахуванням дефектів різних типів // Вісник Житомирського державного технологічного університету. – 2004. – т.2, №4. – С. 180–187.
110. Штовба С.Д., Козачко О.М. Генетична мінімізація вартості контролів в технологічному процесі з урахуванням дефектів різних типів // Вісник Вінницького політехнічного інституту. – 2005. – №3. – С.74–79.
111. Шуренок В.А. Оцінка ефективності застосування СППР на основі теорії нечіткого вірогідностного графу та критеріїв ідентифікації // Вісник Житомирського державного технологічного університету. – 2004 – №4, т. 2. – С. 188–196.
112. Язенин А.В. Квазиэффективные решения задачи многокритериальной нечеткой оптимизации // Изв. АН СССР. Техн. кибернетика. – 1992. – №5. – С. 163–170.
113. Язенин А.В. Модели возможностного программирования в оптимизации систем // Изв. АН СССР. Техн. кибернетика. – 1991. – №5. – С. 133–142.
114. Яншин А.А. Теоретические основы конструирования, технологии и надежности ЭВА: Учеб. пособие для вузов.–М.: Радио и связь, 1983. – 312 с.
115. Яхьева Г.Э. Нечеткие множества и нейронные сети. – М.: Интернет-Университет Информационных Технологий; Бинوم. Лаборатория знаний. – 2006. – 316 с.
116. Acan A. GAACO: A GA + ACO Hybrid for Faster and Better Search Capability. In Proc. of the Third International Workshop on Ant Algorithms «ANTS 2002». Lectures Notes in Computer Science. Vol. 2463. Springer-Verlag. – 2002. – P.300–301.
117. Alaya I., Solnon C., Ghedira K. Ant Algorithm for the Multidimensional Knapsack Problem // Proc. of Inter. Conference on Bioinspired Optimization Methods and their Applications “BIOMA–2004”. – Ljubljana (Slovenia). – 2004. – P.63–72.
118. Ballou D.P., Pazer H.L. Process Improvement Versus Enhanced Inspection in Optimized Systems // International Journal of Production

119. Ballou D.P., Pazer H.L. The Impact of Inspector Fallibility on the Inspection Policy in Serial Production Systems // Management Science. – 1982. – Vol. 28. – P.387–399.
120. Banzhaf W., Nordin P., Keller R.E., Francone F.D. Genetic Programming – An Introduction. Morgan Kaufmann Publishers Inc. –1998.
121. Bellman R.E., Zadeh L.A. Decision–Making in Fuzzy Environment // Management Science. – 1970. –Vol. 17, №4. – P.141–160.
122. Biggs N. L. Discrete Mathematics. Oxford: Oxford University Press, 2002. – 425p.
123. Bonavear E., Dorigo M. Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, 1999. – 307 p.
124. Cai K.-Y. Introduction to Fuzzy Reliability. Dordrecht: Kluwer Academic Publishers. – 1996.
125. Chen J., Yao D.D., Zheng S. Quality Control for Products Supplied with Warranty // Operations Research. – 1998. – Vol. 46. – P. 107–115.
126. Computational Intelligence in Reliability Engineering / Ed. G. Levitin. – Springer, 2007. – 398 p.
127. Cordon O., Herrera F., Stutzle T. A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends // Matheware & Soft Computing. –2002. – №9.
128. Deliman N.C., Feldman R.M. Optimization of Process Improvement and Inspection Location for Serial Manufacturing // International Journal of Production Research. – 1996. – Vol. 34. – P.395–405.
129. Dorigo M., Stutzle T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In Handbook of Metaheuristics (Eds. Glover F and Kochenberger G.). Norwell, MA: Kluwer Academic Publishers. – 2002.
130. Ecisner H.A. Generalized Network Approach to the Planning and Scheduling of a Research Project // Operation Research. – 1962. – P. 116–124.
131. Elmaghraby S. An Algebra for the Analysis of Generalized Activity Networks // Manag. Science. – 1964. – P. 494–513.
132. Eppen G.D., Hurst E.G. Optimal Location of Inspection Stations in a Multistage Production Process // Management Science. – 1974. – Vol. 20. – P. 1194–1200.
133. Fidanova S. Ant Colony Optimization and Multiple Knapsack Problem. In «Handbook of Research on Nature Inspired Computing for Economics and Management» (Ed. Renard J-Ph.). Idea Group Inc., 2006. – P. 498–509.
134. Foster J.W., Malave C.O., Villalobos J.R. Flexible Inspection within

- an Aggregated Information Environment // Computers and Industrial Engineering. – 1990. – Vol. 19. – P. 224–228.
135. Fuzzy Logic Toolbox. User's Guide, Version 2.–The MathWorks, Inc., 1999.
  136. Garcia-Diaz A., Foster J.W., Bonyuet M. Dynamic Programming Analysis of Special Multi-Stage Inspection Systems // IIE Transactions. – 1984. – Vol. 16. – P. 115–125.
  137. Garey M., Johnson D. Computers and Intractability. A Guide to the Theory of NP-Completeness. –New York: W.H. Freeman and Company. – 2000. – 340 p.
  138. Garcia-Diaz A., Foster J.W., Bonyuet M. Dynamic Programming Analysis of Special Multi-Stage Inspection Systems // IIE Transactions. – 1984. – Vol. 16. – P. 115–125.
  139. Gen M., Cheng R. Genetic Algorithms and Engineering Design. – John Wiley & Sons, 1997. – 352 p.
  140. Gunter S.I., Swanson L.A., Inspector Location in Convergent Production Line // International Journal of Production Research. – 1985. – Vol. 23. – P. 1153–1169.
  141. Gunther R.R. An Efficient Evolutionary Algorithm for the Degree-Constrained Minimum Spanning Tree Problem // in Proc. of 14-th ACM Symposium on Applied Computing, San Antonio, TX, 1999. – P.596–603.
  142. Gunther R.R. An Improved Genetic Algorithm for the Multiconstrained 0-1 Knapsack Problem // in Proc. of the 1999 IEEE Congress on Evolutionary Computation, Washington, DC, 1999. –P.291–296.
  143. Gurnani H., Drezner Z., Akella R. Capacity Planning under Different Inspection Strategies // European Journal of Operations Research. – 1996. – Vol. 89. – P. 302–312.
  144. Haupt R., Haupt L. Practical Genetic Algorithms. 2<sup>nd</sup> ed. New Jersey: John Wiley & Sons. – 2004.– 253 p.
  145. Holland J.H. Adaptation in Natural and Artificial Systems. Cambridge: MIT Press. – 1975.
  146. Hsu J.I.S. A Hybrid Inspection System for the Multistage Production Process // International Journal of Production Research. – 1984. – Vol. 22. – P. 63–69.
  147. Hung S.L. Machine Learning: Neural Networks, Genetic Algorithms and Fuzzy Systems. – John Wiley & Sons, 1995.–208 p
  148. Jewkes E.M. Optimal Inspection Effort and Scheduling for a Manufacturing Process with Repair // European Journal of Operational Research. – 1995. – Vol. 85. – P. 340–351.
  149. Jum B.I., Dowell E.D. The Optimal Allocation of Inspection Effort in a Class of Nonserial Production Systems // AIIE Transactions. – 1981.

- P.285–293.
150. Kang K., Ebeling K.A., La S. The Optimal Location of Inspection Stations Using a Rule-Based Methodology // Computers and Industrial Engineering. – 1990. – Vol. 19. – P. 272–275.
  151. Koza J. Genetic Programming - On the Programming of Computers by Means of Natural Selection. – The MIT Press. – 1992.
  152. Leuzamon G., Michalevich Z. A New Version of the Ant System for Subset Problems. In Proc. of Int. Conf. on Evolutionary Computations. – Washington (USA). – 1999. – P. 1459–1464.
  153. Lubics M. On the problem of optimization of a quality inspection process structure // International Journal of Production Research. – 1983. – Vol. 21, No 3. – P.369–381.
  154. Martello S., Toth P. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons. – 1990. – 296 p.
  155. Mehrad S., Hendtlass T. A Fast evaluation strategy for evolutionary algorithms // Applied Soft Computing. – 2003. – №2. – P.156–173.
  156. Mitchell M. Introduction of Genetic Algorithms. Oxford University Press, 2002. – 130 p.
  157. Narahari Y., Khan L.M. Modeling Reentrant Manufacturing Systems with Inspection Stations // Journal of Manufacturing Systems. – 1996. – Vol. 15. – P. 367–378.
  158. Optimization Toolbox. User's Guide, Version 2. The MathWorks, Inc., 1999.
  159. Pilat M., White T. Using Genetic Algorithm to Optimize ACS-TSP. In Proc. of the Third International Workshop on Ant Algorithms «ANTS 2002». Lectures Notes in Computer Science. Vol. 2463. Springer-Verlag. – 2002. – P.282–287.
  160. Rabinowitz G., Emmons H. Optimal and Heuristic Inspection Schedules for Multistage Production Systems // IIE Transactions. – 1997. – Vol. 29. P. 1063–1071.
  161. Raz T., Kaspi M. Location and Sequencing of Imperfect Inspection Operations in Serial Multi-Stage Production Systems // International Journal of Production Research. – 1991. – Vol. 29. – P. 1645–1659.
  162. Reimann M., Shtovba S., Nepomuceno E. A Hybrid Ant Colony Optimization and Genetic Algorithm Approach for Vehicle Routine Problems Solving // Student Papers of Complex Systems Summer School – 2001, Hungary, Budapest, P.134-141.
  163. Rotshtein A. Fuzzy Models of Labour Systems Functioning Reliability and Quality // Ergonomics of Hybrid Automated Systems, Elsevier, 1992, pp. 139-145.
  164. Rotshtein A. Fuzzy Reliability Analysis of Man–Machine Systems. In «Reliability and Safety Analysis under Fuzziness. Studies in

- Fuzziness», Vol. 4, Phisica-Verlag, A Springer –Verlag Company, 1994. – P. 245–270.
165. Rotshtein A. Design and Tuning of Fuzzy Rule-Based System for Medical Diagnosis. In “Fuzzy and Neuro-Fuzzy Systems in Medicine” (Eds.: Teodorescu N.H., Kandel A., and Jain L.C.). Boca-Raton: CRC-Press, 1998. – P. 243–289.
  166. Rotshtein A., Shtovba S. Assessment and Optimization of Reliability of Algorithmic Processes under Fuzziness // Proc. of the 12<sup>th</sup> Industrial Engineering and Management Conference, Tel-Aviv (Israel), 2002. – P. 379–383.
  167. Rotshtein A., Shtovba S., Posner M. Optimization of a Water Purification Technological Process by Genetic Algorithms // Proc. of the Third Workshop on Binding Environmental Science and Artificial Intelligence in frame of European Conference on Artificial Intelligence (ECAI-2002), Lyon (France), 2002. – P. 67–70.
  168. Rotshtein A., Shtovba S. Fuzzy Probability –Based Modeling the Reliability of Algorithmic Processes. In Proc. of International Symposium on Stochastic Models in Reliability, Safety, Security and Logistics. Beer Sheva (Israel), 2005. – P. 300–303.
  169. Rotshtein A., Shtovba S. Genetic Optimization of Multidimensional Technological Process Reliability. In «Computational Intelligence in Reliability Engineering: Evolutionary Techniques in Reliability Analysis and Optimization». Studies in Computational Intelligence, vol. 39 (Ed. G. Levitin). – 2007. P. 287–300.
  170. Rotshtein A., Shtovba S. Modeling of Algorithmic Process Reliability with Fuzzy Source Data // Eksploatacja i niezawodnosc (Maintenance and Reliability). – 2006. – № 2. – P.40–43.
  171. Rotshtein A., Shtovba S. Risk Assessing by Fuzzy Logic-Algorithmic Fault Tree / Труды Пятой межд. научной школы «Моделирование и анализ безопасности и риска в сложных системах». – СПб, 2005. – С. 339–343.
  172. Rotshtein, A. and Shtovba, S. Fuzzy Reliability Analysis and Optimization of Algorithmic Processes. In Proc. of the Fifth European Congress on Intelligent Techniques and Soft Computing, Aachen (Germany). – 1997. – P. 67–71.
  173. Shin W.S., Hart S.M., Lee H.F. Strategic Allocation of Inspection Stations for a flow Assembly line: a Hybrid Procedure // IIE Transactions. – 1995. – Vol. 27. – P. 707–715.
  174. Solnon C., Bridge D. An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems. In «System Engineering using Particle Swarm Optimization» (Eds.: Nedjah N. and Mourelle L.). Nova Science Publisher, 2006. – P. 7–29.

175. Stutzle T., Hoos H.H. MAX-MIN Ant System // Future Generation Computer Systems. – 2000. – №8 (16). – P.889–914.
176. Tang C.S. Designing an Optimal Production System with Inspection // European Journal of Operations Research. – 1991. – Vol. 52. – P. 45–54.
177. Villalobos J.R., Foster J.W. Some Results from Model of Dynamic Inspection Allocation // Computers and Industrial Engineering. – 1991. – Vol. 21. – P. 355–358.
178. Villalobos J.R., Foster J.W., Disney R.L. Flexible Inspection Systems for Serial Multi-Stage Production Systems // IIE Transactions. – 1993. – Vol. 25. – P. 16–26.
179. Whitehouse G.E. System Analysis and Design Using Network Techniques. –Prentice-Hall, Inc. Englewood Cliffs, 1973. –243p.
180. Yager R.R. Fuzzy Decision Making Including Unequal Objectives // Fuzzy Sets and Systems. – 1978. – №1. – P. 87–95.
181. Yao D.D., Zheng S. Sequential Inspection under Capacity Constraints // Operations Research. – 1999. – Vol. 47. – P. 410–422.
182. Yi-Liang Xu, Meng-Hiot Lim, Yew-Soon Ong, Jing Tang A GA-ACO-Local Search Hybrid Algorithm for Solving Quadratic Assignment Problem. In Proc. of the 8th Conference on Genetic and Evolutionary Computation «GECCO'06», Seattle (USA), 2006. – P. 599–605.
183. Yum B.J., McDowell E.D. Optimal Inspection Policies in a Serial Production System Including Scrap Rework and Repair: an MILP Approach // International Journal of Production Research. – 1987. – Vol. 25. – P. 1451–1464.
184. Zadeh L. Fuzzy Sets // Information and Control. – 1965. – №8. – P. 338–353.
185. Zadeh L. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes // IEEE Trans. Syst. Man Cybernet. – 1973. – №3. – P. 28–44.
186. Zimmerman H.-J. Fuzzy Sets Theory and Its Applications. 3<sup>rd</sup> ed. – Kluwer Academic Publisher, 1996. – 435 p.

## Додаток А

Опис реквізитів форми платіжного доручення  
В додатку використовуються матеріали з [3].

- |  |  |
|--|--|
| 1 – Номер документа  | 22 – Кор, рахунок отримувача               |
| 2 – Дата документа   | 23 – ІНН отримувача                        |
| 3 – Спосіб відправки                                       | 24 – КПП отримувача                        |
| 4 – Показник статусу                                       | 25 – Форма власності отримувача            |
| 5 – ІНН  | 26 – Назва отримувача                      |
| 6 – КПП  | 27 – Рахунок отримувача                    |
| 7 – Сума   | 28 – Пошук                                 |
| 8 – Форма власності платника                               | 29 – ПДВ                                   |
| 9 – Платник  | 30 – Вид операції                          |
| 10 – Рахунок платника                                      | 31 – Черговість платежу                    |
| 11 – Тип банку платника                                    | 32 – Ставка ПДВ у відсотках                |
| 12 – Назва банку платника                                  | 33 – Спосіб обчислення ПДВ                 |
| 13 – БІК або код платника                                  | 34 – Код бюджетної класифікації            |
| 14 – Коротка назва типу населеного пункту банку платника   | 35 – Код ОКАТО                             |
| 15 – Назва населеного пункту банку платника                | 36 – Показник призначення платежу          |
| 16 – Кореспондентський рахунок платника                    | 37, 38, 39 – Показники податкового періоду |
| 17 – Тип банку отримувача                                  | 40 – Показник номера документа             |
| 18 – Назва банку отримувача                                | 41, 42, 43 – Показник дати документа       |
| 19 – БІК або код отримувача                                | 44 – Показник типу платежу                 |
| 20 – Коротка назва типу населеного пункту банку отримувача | 45 – Призначення платежу                   |
| 21 – Назва населеного пункту банку отримувача              | 46 – Код типу валютної операції            |
|  | 47 – Номер паспорта угоди                  |



## Додаток Б

### Класифікація помилок в системі «клієнт-банк»

В додатку використовуються матеріали з [3].

#### 1. Форматні помилки

- 1.1. Відсутній номер документа;
- 1.2. Незаповнена підстава платежу;
- 1.3. В платежу не вказано показник статусу платника податку;
- 1.4. Не вказано значення КБК;
- 1.5. Не вказано значення коду ОКАТО;
- 1.6. Не вказано «Показник підстави платежу»;
- 1.7. Не вказано «Показник податкового періоду»;
- 1.8. Не вказано КПП платника;
- 1.9. Не вказано КПП отримувача;
- 1.10. Не вказано «Показник номера документа»;
- 1.11. Поле «Показник дати документа» некоректно заповнено;
- 1.12. Не вказано показник типу платника;
- 1.13. Не вказано ІНН платника;
- 1.14. Не вказано ІНН отримувача;
- 1.15. Значення поля «Показник статусу» повинен бути обраний з довідника;
- 1.16. Номер документа перевищує допустиму довжину;
- 1.17. При розрахунку з участю нерезидентів в підставі платежу повинна бути вказана інформація про вид валютної операції та паспорт угоди при його наявності;
- 1.18. Некоректний формат інформації про код типу валютної операції та паспорт угоди;
- 1.19. Наявність недопустимих символів в паспорті угоди;
- 1.20. Неправильна довжина рахунку отримувача;
- 1.21. Неправильний ключ рахунку отримувача;
- 1.22. Номер документа містить пробіли;
- 1.23. Номер документа містить недопустимі символи;
- 1.24. Номер документа дорівнює нулю;
- 1.25. Недопустима рівність нулю трьох останніх цифр номера документа;
- 1.26. Номер документа перевищує три символи;
- 1.27. Рахунок платника повинен мати довжину з 20 символів;
- 1.28. Ключ рахунку платника неправильний;
- 1.29. Неправильна довжина рахунку отримувача (20 символів);
- 1.30. В документі вказана нульова або від'ємна сума;

- 1.31. Довжина підстави платежу перевищує допустиму;
- 1.32. В підставі платежу не знайдено згадування про ПДВ;
- 1.33. В платежу виявлено більше одного згадування про ПДВ;
- 1.34. Некоректне значення в полі «Показник податкового періоду»
- 1.35. Некоректне значення «Показник податкового періоду» для типу періодичності;
- 1.34. Показник дати документа для документів з підставою платежу повинен вказуватися обов'язково;
- 1.37. Неправильне значення в полі «Показник дати документа»;
- 1.38. Неправильна довжина КПП платника;
- 1.39. Неправильна довжина КПП отримувача;
- 1.40. Наявність недопустимих символів в полі КПП отримувача;
- 1.41. Поле «ІНН платника» містить недопустимі символи;
- 1.42. Неправильна довжина ІНН платника, Допустимо 10 символів у юридичних осіб та 12 символів у фізичних осіб;
- 1.43. Поле «ІНН отримувача» містить недопустимі символи.

## **2. Операційні помилки**

- 2.1. Вказаний код валютної операції відсутній в довіднику кодів;
- 2.2. Валюта рахунку отримувача не знайдена в довіднику валют;
- 2.3. Рахунок платника # закритий для платежів;
- 2.4. БИК отримувача відсутній в файлах NSI;
- 2.5. Недопустимий для платіжного доручення тип операції;
- 2.6. Неправильний спосіб відправки;
- 2.7. Введене значення коду бюджетної класифікації відсутнє в довіднику;
- 2.8. Значення поля «Показник підстави платежу» повинне бути в довіднику;
- 2.9. Введене значення показника типу платежу відсутнє в довіднику;
- 2.10. Код валюти рахунку отримувача не відповідає коду національної валюти;
- 2.11. Дата документа раніша від допустимої;
- 2.12. Дата документа перевищує допустиму;
- 2.13. Документ з таким номером на цю дату вже існує;
- 2.14. Рахунок платника йому не належить;
- 2.15. Код валюти рахунку платника не відповідає коду національної валюти;

- 2.16. ІНН платника в документі не відповідає вказаному в довіднику рахунків для цього рахунку;
- 2.17. КПП платника, що вказаний в документі, не відповідає вказаному в реквізитах;
- 2.18. Рахунок отримувача повинен відрізнятися від рахунку платника;
- 2.19. Дані про банк платника не збігаються з довідником банків;
- 2.20. Код валюти рахунку отримувача не відповідає коду національної валюти;
- 2.21. Дані про банк отримувача не збігаються з довідником банків;
- 2.22. Документ з таким номером вже існує;
- 2.23. Документ № %0:s від %1:s на суму %2:s валюти %3:s з рахунком %4:s в банку %5:s на рахунку %6:s вже існує;
- 2.24. Виконання документа може призвести до нестачі коштів на рахунку;
- 2.25. Вказана в документі сума перевищує допустимий ліміт;
- 2.26. Сума документів, які утворені за останні %кілька% днів, включаючи поточний, перевищує допустимий ліміт.

### **3. Помилки електронно-цифрового підпису**

- 3.1. Неправильний електронно-цифровий підпис;
- 3.2. Електронно-цифровий підпис неправомірний.

## Додаток В

Початкові дані для прогнозування часу проведення платіжних доручень в системі «клієнт-банк»

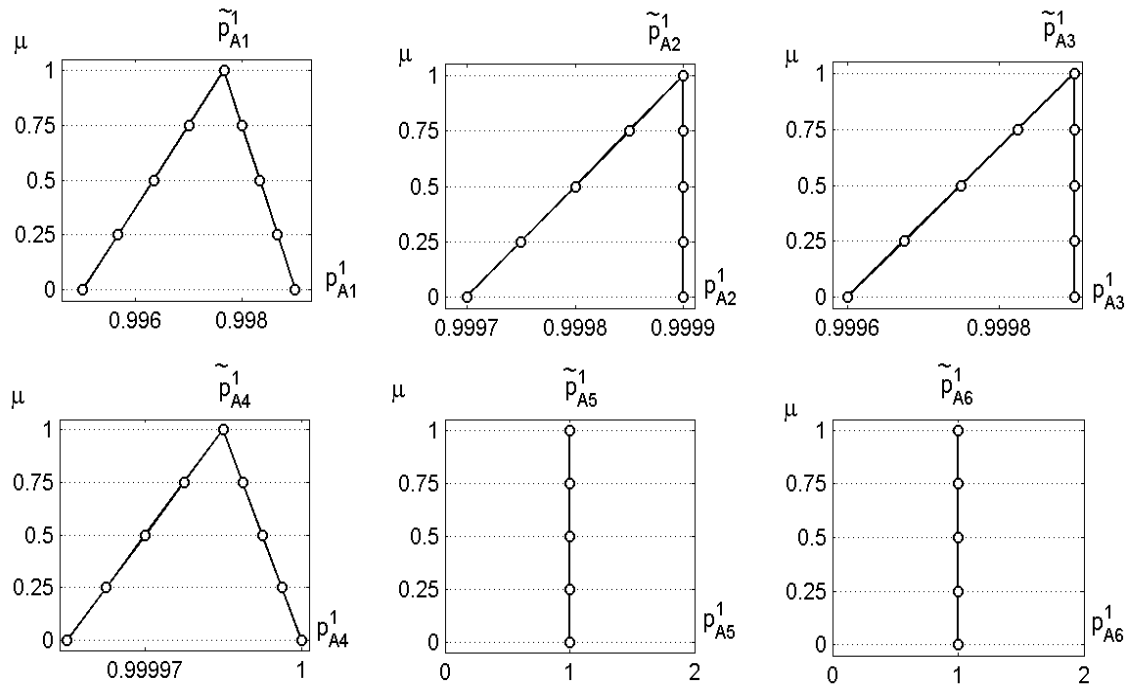


Рис. В.1. Функції належності  $p_{A_i}^1$ ,  $i = \overline{1,6}$

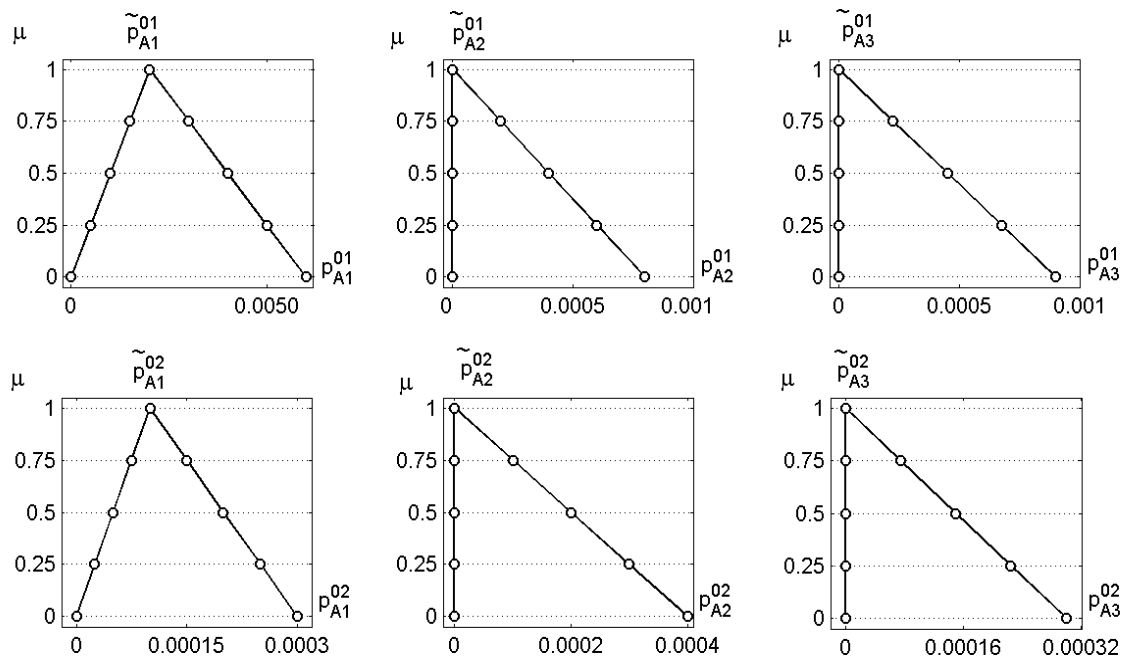


Рис. В.2. Функції належності  $p_{A_i}^{01}$  та  $p_{A_i}^{02}$ ,  $i = \overline{1,3}$

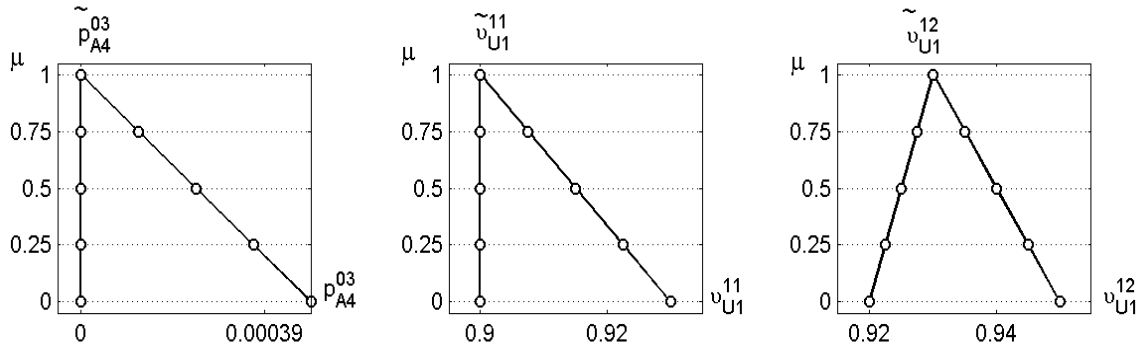


Рис. В.3. Функції належності  $p_{A_4}^{0_3}$ ,  $v_{U_1}^{1_1}$ ,  $v_{U_1}^{1_2}$

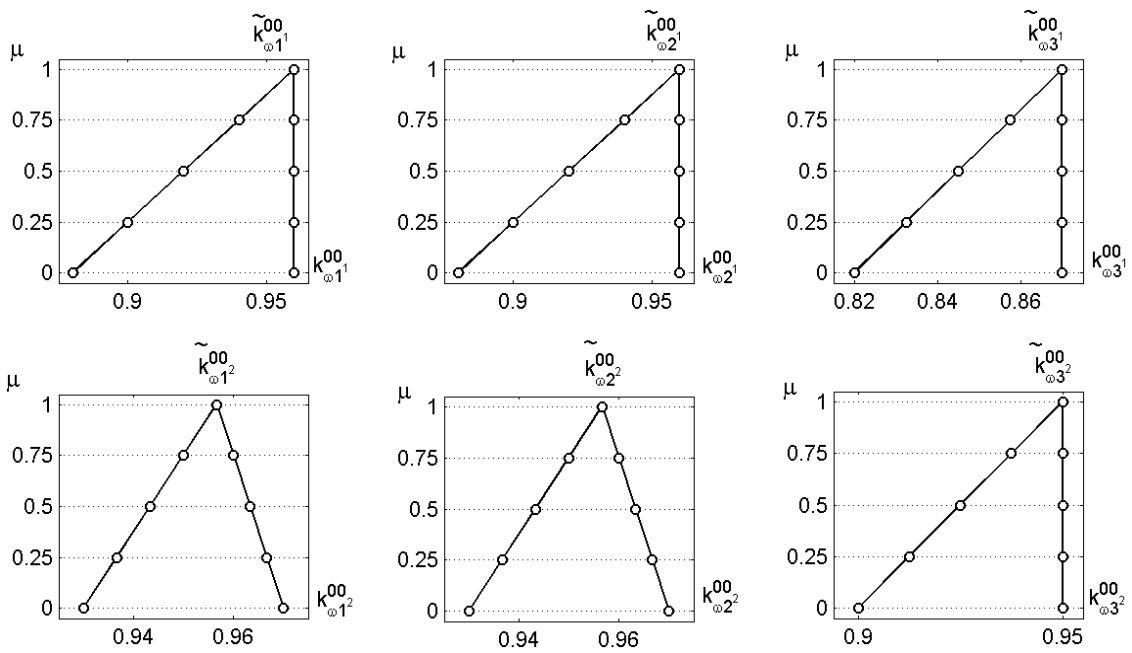


Рис. В.4. Функції належності  $k_{\omega_i}^{00_1}$  та  $k_{\omega_i}^{00_2}$ ,  $i = \overline{1,3}$

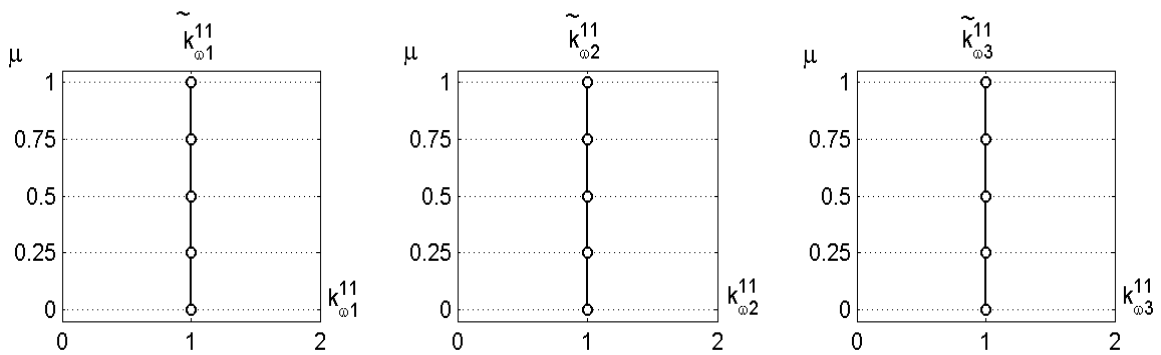


Рис. В.5. Функції належності  $k_{\omega_i}^{11}$ ,  $i = \overline{1,3}$

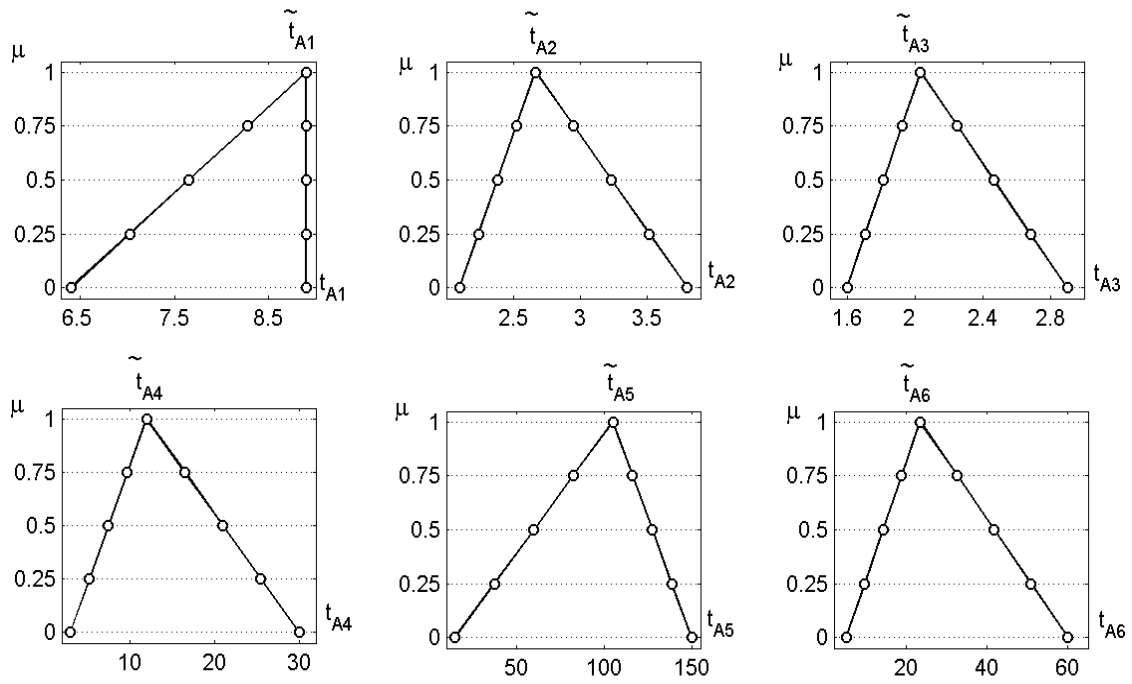


Рис. В.6. Функції належності  $t_{A_i}$ ,  $i = \overline{1, 6}$

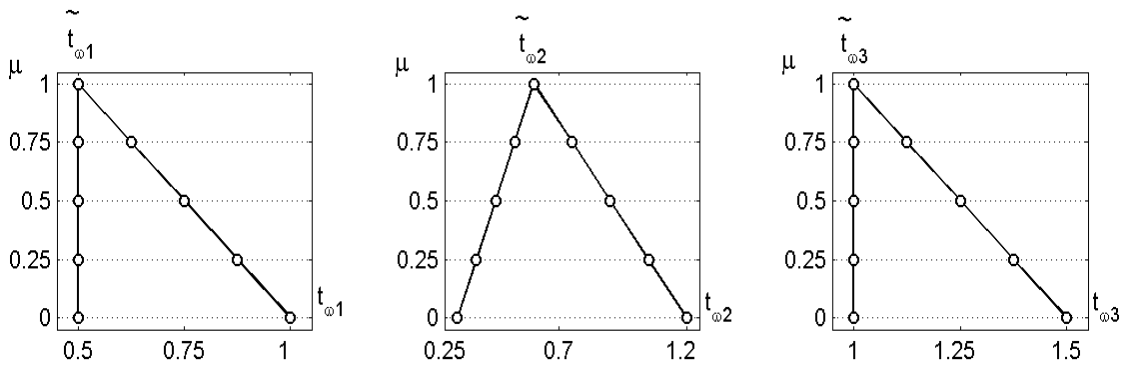


Рис. В.7. Функції належності  $t_{\omega_i}$ ,  $i = \overline{1, 3}$

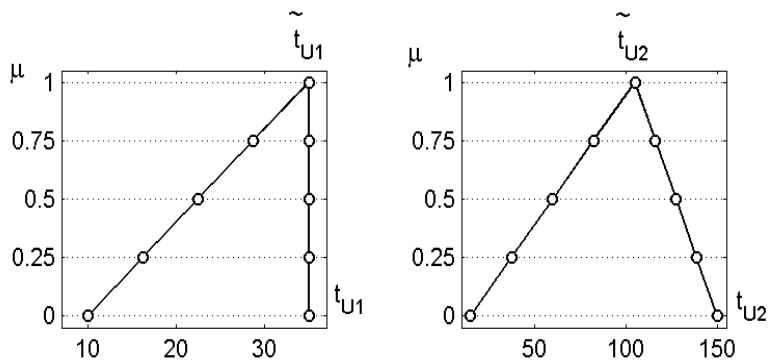


Рис. В.8. Функції належності  $t_{U_1}$  та  $t_{U_2}$

## Додаток Г

### Покроковий аналіз надійності системи «клієнт-банк»

$B1=A1^N1$	– «багаторазова робота»
$B2=A2^N2$	– «багаторазова робота»
$B3=A3^N3$	– «багаторазова робота»
$B4=B1 \& B2$	– «композиція»
$B5=B3 \& B4$	– «композиція»
$B6=\omega1-\{U1\}$	– « $\omega$ -добробка»
$B7=B5 \& B6$	– «композиція»
$B8=B7 \& A4$	– «композиція»
$B9=B8 \& A5$	– «композиція»
$B10=U2 \& B6$	– «композиція»
$B11=B10 \& A4$	– «композиція»
$B12=B11 \& A5$	– «композиція»
$B13=\{B12\}-\omega2$	– « $\omega$ -ітерація»
$B14=B9-\omega2-B13$	– «робота–контроль–добробка»
$B15=B14 \& A6$	– «композиція»
$B16=B12-\omega2-B13$	– «робота–контроль–добробка»
$B17=\{B16\}-\omega3$	– « $\omega$ -ітерація»
$B18=B15-\omega3-B17$	– «робота–контроль–добробка»

### Покроковий аналіз АП

Крок 1 | Оператор В1:

Альфа-рівень	Час виконання	
0,0000	24,0000	36,0000
0,2500	27,0000	36,0000
0,5000	30,0000	36,0000
0,7500	33,0000	36,0000
1,0000	36,0000	36,0000

Крок 2 | Оператор В2:

Альфа-рівень	Час виконання	
0,0000	8,0000	16,0000
0,2500	8,6667	14,6667
0,5000	9,3333	13,3333
0,7500	10,0000	12,0000
1,0000	10,6667	10,6667

Крок 3 | Оператор В3:

Альфа-рівень	Час виконання	
	0,0000	6,0000
0,2500	6,5000	11,0000
0,5000	7,0000	10,0000
0,7500	7,5000	9,0000
1,0000	8,0000	8,0000

Крок 4 | Оператор В4:

Альфа-рівень	Час виконання	
	0,0000	32,0000
0,2500	35,6667	50,6667
0,5000	39,3333	49,3333
0,7500	43,2000	48,0000
1,0000	46,6667	46,6667

Крок 5 | Оператор В5:

Альфа-рівень	Час виконання	
	0,0000	38,0000
0,2500	42,1667	61,6667
0,5000	46,3333	59,3333
0,7500	50,5000	57,0000
1,0000	54,6667	54,6667

Крок 6 | Оператор В6:

Альфа-рівень	Час виконання	
	0,0000	11,0000
0,2500	17,2816	37,7469
0,5000	23,5888	37,2550
0,7500	29,9236	36,7686
1,0000	36,2875	36,2875

Крок 7 | Оператор В7:

Альфа-рівень	Час виконання	
	0,0000	49,0000
0,2500	59,4482	99,7974
0,5000	69,9222	96,8892
0,7500	80,4236	93,9898
1,0000	90,9542	90,9542



Крок 8 | Оператор В8:

Альфа-рівень	Час виконання	
0,0000	52,0000	132,7144
0,2500	64,6982	125,2974
0,5000	77,4222	117,8892
0,7500	90,1736	110,4898
1,0000	102,9542	102,9542

Крок 9 | Оператор В9:

Альфа-рівень	Час виконання	
0,0000	67,0000	282,7144
0,2500	102,1982	264,0474
0,5000	137,4222	245,3892
0,7500	172,6736	226,7398
1,0000	208,0987	208,0987

Крок 10 | Оператор В10:

Альфа-рівень	Час виконання	
0,0000	36,0000	187,1334
0,2500	63,0846	176,4258
0,5000	90,1703	165,7249
0,7500	117,2573	155,0314
1,0000	144,3455	144,3455

Крок 11 | Оператор В11:

Альфа-рівень	Час виконання	
0,0000	39,0000	217,1334
0,2500	68,3346	201,9258
0,5000	97,6703	186,7249
0,7500	127,0073	171,5314
1,0000	156,3455	156,3455

Крок 12 | Оператор В12:

Альфа-рівень	Час виконання	
0,0000	54,0000	367,1334
0,2500	105,8346	340,6758
0,5000	157,6703	314,2249
0,7500	209,5073	287,7814
1,0000	261,3455	261,3455

Крок 13 | Оператор В13:

Альфа-рівень	Час виконання	
0,0000	56,0000	372,2558
0,2500	108,0847	345,2503
0,5000	160,1706	318,2654
0,7500	212,2579	291,2984
1,0000	264,3470	264,3470

Крок 14 | Оператор В14:

Альфа-рівень	Час виконання	
0,0000	70,4000	286,2713
0,2500	105,0738	266,6252
0,5000	139,7918	247,2872
0,7500	174,5576	228,2171
1,0000	209,3749	209,3749

Крок 15 | Оператор В15:

Альфа-рівень	Час виконання	
0,0000	75,4000	346,2713
0,2500	114,6571	317,4586
0,5000	153,9585	288,9539
0,7500	193,3076	260,7171
1,0000	232,7082	232,7082

Крок 16 | Оператор В16:

Альфа-рівень	Час виконання	
0,0000	55,0000	369,5997
0,2500	106,9584	342,8883
0,5000	158,9169	316,1943
0,7500	210,8754	289,5110
1,0000	262,8341	262,8341

Крок 17 | Оператор В17:

Альфа-рівень	Час виконання	
0,0000	62,0000	432,6027
0,2500	118,5417	396,4728
0,5000	175,0836	360,3612
0,7500	231,6254	324,2611
1,0000	288,1674	288,1674

Крок 18 | Оператор В18:

Альфа-рівень	Час виконання	
0,0000	76,4000	348,2020
0,2500	115,6579	319,0543
0,5000	154,9607	290,3021
0,7500	194,3122	261,8769
1,0000	233,7160	233,7160

*Наукове видання*

**Ротштейн Олександр Петрович  
Штовба Сергій Дмитрович  
Козачко Олексій Миколайович**

**МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ НАДІЙНОСТІ  
БАГАТОВИМІРНИХ АЛГОРИТМІЧНИХ ПРОЦЕСІВ**

*Монографія*

Редактор С. Малішевська

Оригінал-макет підготували С. Штовба та О. Козачко

Видавництво ВНТУ «УНІВЕРСУМ-Вінниця»

Свідоцтво Держкомінформу України

серія ДК № 746 від 25.12.2001 р.

21021, м. Вінниця, Хмельницьке шосе, 95

ВНТУ, ГНК, к. 114

Тел. (0432) 59-85-32

Підписано до друку 15.11.2007 р.

Формат 29,7x42¼ Папір офсетний

Гарнітура Times New Roman

Друк різнографічний Ум. др. арк. 12,19

Наклад 100 прим. Зам № 2007-164

Віддруковано в комп'ютерному інформаційно-видавничому центрі

Вінницького національного технічного університету

Свідоцтво Держкомінформу України

серія ДК № 746 від 25.12.2001 р.

21021, м. Вінниця, Хмельницьке шосе, 95

ВНТУ, ГНК, к. 114

Тел (0432) 59-81-59