

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

ІНСТИТУТ ІНФОРМАТИКИ ТА РАДІОЕЛЕКТРОНІКИ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК І ТЕХНОЛОГІЙ
(повне найменування інституту, факультету)

КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ОБЧИСЛЮВАЛЬНОЇ
МАТЕМАТИКИ
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему Задача про надійність з нормально розподіленими вхідними
параметрами

Виконав: студентка 4 курсу, групи КНТ – 827сп

Спеціальності 124 – Системний аналіз
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інтелектуальні технології та прийняття рішень
в складних системах

Залата Ю.О.

(прізвище та ініціали)

Керівник Подковаліхіна О.О.

(прізвище та ініціали)

Рецензент Стеганняв Є.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повно найменування закладу вищої освіти)

Інститут Інформатики та радіоелектроніки
 Факультет Комп'ютерних наук і технологій
 Кафедра Системного аналізу та обчислювальної математики
 Ступінь вищої освіти бакалавр
 Спеціальність 124 – Системний аналіз
(код – найменування)
 Освітня програма (спеціалізація) Інтелектуальні технології та прийняття рішень
в складних системах
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

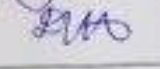
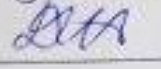
 **Завідувач кафедри проф. Г.В. Корнієв**
 « 04 » червня 2020 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Залати Юлії Олександрівни
(прізвище, ім'я, по батькові)

- Тема проєкту (роботи) Задача про надійність з нормально розподіленими вхідними параметрами
 керівник проєкту (роботи) доцент Подковаляхіна Олена Олександрівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
 затверджені наказом закладу вищої освіти від «19» травня 2020 року № 136
- Строк подання студентом проєкту (роботи) 04 червня 2020 року
- Вихідні дані до проєкту (роботи) умова задачі про надійність, перелік літературних джерел згідно теми дослідження
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Проведено аналіз останніх публікацій та досліджень за темою роботи. Розроблено програмну реалізацію для розв'язання задачі про надійність з нормально розподіленими параметрами методом перебору даних, що розраховує оптимальний план та значення надійності, і заносить дані в Excel. Отримано та проаналізовано розв'язок задачі з вхідними параметрами з 10%, 20%, 30% відхиленнями для різних критеріїв оптимальності.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	доцент Подковаліхіна О.О		
2	доцент Подковаліхіна О.О		
Нормоконтроль	ст. викл. Широкопад Д.В.		

7. Дата видачі завдання « 10 » жовтня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Сформулювати мету та основні завдання дипломної роботи.	10.10.2019-30.10.2019	
2	Опрацювати літературу та існуючі дослідження за темою роботи.	01.11.2019-15.11.2019	
3	Оптимізація програми для вирішення задачі.	16.11.2019-25.11.2019	
4	Розрахунки та аналіз даних.	12.02.2020-05.05.2020	
5	Оформлення пояснювальної записки.	06.05.2020-16.05.2020	
6	Попередній захист дипломного проєкту та отримання рецензій.	26.05.2020	
7	Захист дипломної роботи.	04.06.2020	

Студентка


(підпис)

Залата Ю.О.
(прізвище та ініціали)

Керівник проєкту (роботи)


(підпис)

Подковаліхіна О.О.
(прізвище та ініціали)

РЕФЕРАТ

Дипломна робота: 40 с., 8 рис., 4 табл., 1 дод., 16 літературних джерел.

Об'єкт дослідження – моделі динамічного програмування в умовах статистичної невизначеності.

Предмет дослідження – задача про надійність в умовах статистичної невизначеності.

Мета роботи – аналіз оптимального розв'язку задачі про надійність з нормально розподіленими параметрами.

Методи дослідження – метод повного перебору.

В дипломній роботі розроблено програмну реалізацію для розв'язання задачі про надійність з нормально розподіленими параметрами методом перебору даних, що розраховує оптимальний план та значення надійності, і заносить дані в Excel. Отримано та проаналізовано розв'язок задачі для різних критеріїв оптимальності.

МОДЕЛІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ, ЗАДАЧА ПРО НАДІЙНІСТЬ З НОРМАЛЬНО РОЗПОДІЛЕНИМИ ПАРАМЕТРАМИ, МЕТОД ПЕРЕБОРУ ДАНИХ, СТАТИСТИЧНА НЕВИЗНАЧЕНІСТЬ, КРИТЕРІЇ ОПТИМАЛЬНОСТІ, MATLAB.

ЗМІСТ

Завдання	2
Реферат	4
Вступ.....	6
1 Аналіз останніх досліджень і публікацій.....	7
1.1 Історія задач динамічного програмування	7
1.2 Класичні задачі методу динамічного програмування	8
1.2.1 Задача про рюкзак	8
1.2.2 Задача про розподіл ресурсів	11
1.3 Надійність електронного пристрою	14
1.4 Загальні методи розрахунку надійності.....	17
2 Задача про надійність електронного пристрою	21
2.1 Постановка задачі.....	21
2.2 Програмна реалізація задачі про надійність з нормально розподіле- ними параметрами.....	22
2.3 Результати розрахунків задачі про надійність з нормально розпо- леними параметрами	28
2.4 Аналіз розрахунків задачі задачі про надійність з нормально розпо- діленими параметрами.....	31
Висновок	33
Перелік посилань.....	34
Додаток А. Код реалізації програми	36

ВСТУП

Більшість задач з економіки можна розглядати як задачі з багатокроковою структурою. Це кожен процес перспективного і поточного планування і управління, що розвивається в часі. Кроком в них може бути рік, квартал, місяць, тиждень і т. д. Розв'язок виконується або шляхом укладання комплексу взаємопов'язаних статичних моделей для кожного періоду, або шляхом укладання єдиної динамічної задачі оптимального програмування з застосуванням багатокрокової процедури прийняття рішень.

Властивість технічних об'єктів зберігати, у межах певного часу, значення всіх параметрів, які формулюють можливість виконувати належні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування – називається надійністю.

В теорії надійності вводяться параметри надійності об'єктів, охарактеризовуються норми до надійності з врахуванням економічних та інших факторів, розробляються рекомендації по забезпеченню заданих вимог до надійності на етапах проектування, виробництва, зберігання та експлуатації.

Випробування на надійність проводяться за для того, щоб на початкових стадіях життєвого циклу виробу виявити можливі помилки та мати впевненість в тому, що система буде відповідати заданим вимогам. В наш час, частіше застосовуються прискорені випробування у динамічно змінному середовищі у тому числі і структурно-складних систем з урахуванням їх старіння, втоми, зносу та деградації в ході експлуатації.

1 АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

1.1 Історія задач динамічного програмування

Вперше термін був використаний в 1940 - х роках Річардом Беллманом, за для опису процесів розв'язання задач, в яких потрібно знайти кращі вирішення одне за іншим. Внесок Беллмана запам'ятовується в назві рівняння Беллмана. Даний метод має широке використання в біо-інформатиці для таких задач, як вирівнювання послідовності, згортання білка, передбачення структури РНК і білка-зв'язування ДНК. Перші алгоритми динамічного програмування для зв'язування білок-ДНК були розроблені в 1970 - х роках незалежно один від одного Чарльзом Delisi в США та Георгієм Гурським і Олександром Заседателевим в СРСР. Останнім часом ці алгоритми стали дуже популярними в області біо-інформатики та обчислювальної біології, зокрема, в дослідженнях нуклеосом позиціонування і фактор транскрипції зв'язування [1].

Останнім часом, при синтезі технологічних схем поділу, досить широке використання має метод динамічного програмування. Він полягає в тому, що починаючи з кінця, оптимальні схеми синтезують крок за кроком. В даному випадку, технологічна схема розглядається як багатостадійний процес поділу без зворотніх масових і енергетичних потоків. На початковому етапі розглядаються колони, в яких діляться бінарні суміші, а далі трьох - чотирьох компонентні і т.д., з урахуванням оптимального варіанту на попередньому етапі.

У різних галузях економіки застосовуються моделі динамічного програмування для аналізу, прогнозування й підбору оптимальних рішень. Сюди входить планування виробництва та його оперативне керування, управління трудовими ресурсами та запасами, розподіл ресурсів, планування і розміщення об'єктів, керівництво проектом, розподіл інвестицій тощо [2].

1.2 Класичні задачі методу динамічного програмування

Починаючи з часу створення динамічного програмування, як окремого методу, котрий вимагає спеціальні знання, пройшло багато часу, тому в ньому була сформульована група класичних задач. Звичайно, на практиці зустрічаються і інші завдання, але даний класифікатор відображає основні види, з якими необхідно ознайомитися, щоб мати змогу розібратися в самому методі [3]:

1. Завдання про обчислення чисел Фібоначчі.
2. Завдання про найбільшу загальну підпоследовність.
3. Завдання пошуку найбільшої збільшеної підпоследовності.
4. Завдання про редакційну відстань (відстань Левенштейна).
5. Завдання про порядок множення матриць.
6. Завдання про вибір траєкторії.
7. Завдання послідовного прийняття рішення.
8. Питання про використання робочої сили.
9. Задача про розподіл ресурсів.
10. Задача про ранець.
11. Алгоритм Флойда - Уоршелла: знайти найкоротші відстані між усіма вершинами зваженого орієнтованого графа.
12. Алгоритм Беллмана - Форда: знайти найкоротший шлях в зваженому графі між двома заданими вершинами.

1.2.1 Задача про рюкзак

У загальному вигляді задача про рюкзак (в літературі часто зустрічається і інша її назва, а саме, задача про ранець) формулюється в такий спосіб: є рюкзак певного об'єму і необмежена кількість предметів. Для кожного предме-

та відомі його об'єм (вага) і цінність (вартість, ефективність). У рюкзак можна покласти ціле число предметів різного типу. Мета завдання полягає в тому, щоб отримати максимальну цінність предметів, що знаходяться в рюкзаку, а їх об'єм (вага) не перевищував заданої величини. До подібного формулювання може бути зведена задача максимального використання вантажопідйомності рухомого складу, вантажомісткості судна, автомобіля і т.д. Найчастіше данна задача ставиться при виборі оптимального управління в економіко-фінансових областях (наприклад розподіл бюджету відділу по проектам).

В роботі [4] наведено конструювання і експериментальне дослідження трьох алгоритмів, призначених для розв'язання задачі про рюкзак. Два з них – метод гілок і меж і динамічного програмування - дозволяють отримати точні розв'язки, третій метод – евристичний - дає наближені рішення задачі. Наводяться порівняльні характеристики алгоритмів, на підставі чого даються рекомендації щодо їх практичного застосування.

В роботі [5] розглядається алгоритм точного розв'язання задачі про рюкзак методом динамічного програмування. Ключовою особливістю даного алгоритму є лінійна залежність часової складності алгоритму від розмірності задачі. Дослідження алгоритму динамічного програмування актуально, так як його швидкодію визначає стійкість рюкзачних систем шифрування.

При додатковому обмеженні на вагу предметів, задачу про рюкзак можна вирішити за псевдополіноміальний час методами динамічного програмування. Наприклад, нехай вага кожного предмета W_i є цілим невід'ємним числом. Тоді для вирішення завдання необхідно обчислити оптимальні рішення для всіх $w \in \mathbb{Z} : 0 \leq w \leq W$, де W - задана вантажопідйомність.

Визначимо $m[w]$ як максимальну цінність предметів, які можна помістити в рюкзак вантажопідйомністю w .

Для $m[w]$ можна записати рекурентні формули:

- $m[0] = 0$
- $m[w] = \max_{w_i \leq w} (v_i + m[w - w_i]),$

де v_i, w_i - цінність і вага i -го предмета відповідно, а максимум з порожньої множини слід вважати рівним нулю.

Функціональне рівняння Р. Беллмана (або функціональне рівняння динамічного програмування) - є фактично останнє рівняння. В цьому випадку для його розв'язання достатньо розрахувати кожне значення $m[w]$, починаючи від 0 і до W . Можемо на кожному кроці додатково зберігати набір предметів, що реалізує максимальну цінність, тоді алгоритм видасть і оптимальний набір предметів.

Задля того щоб на кожному кроці можна було знайти максимум з n предметів, алгоритм має обчислювальну складність $O(nW)$. Тому ефективність даного алгоритму визначається значенням W . Алгоритм дає відмінні результати при $W \leq 1000$, але не дуже хороші для $W \geq 10\,000\,000$.

Співвідношення, які являються рекурентними, можна записувати як щодо ваги предметів, так і щодо цінності. Для цього позначимо за $y[i, V]$ мінімальна вага предметів, сумарною цінністю V , який можна отримати з перших i предметів. Якщо необхідну вагу отримати неможливо, то відзначимо це як

$$y[i, V] = W + 1$$

Після цього розв'язуємо функціональне рівняння динамічного програмування.

Алгоритми мурашиних колоній - це нові перспективні методи оптимізації, що базуються на моделюванні поведінки колонії мурах. Колонія мурах може розглядатися як багатоагентна система, в якій кожен агент (мураха) функціонує за дуже простими правилами. На противагу майже примітивній поведінці агентів, поведінка всієї системи виходить напрочуд розумною. За допомогою мурашиних алгоритмів були розв'язані такі задачі комбінаторної оптимізації, як задача комівояжера, квадратична задача про призначення, задача маршрути-

зації вантажівок, задача календарного планування, задача розфарбовування графа, а також розглянуто задачу про рюкзак [6].

1.2.2 Задача про розподіл ресурсів

Метод динамічного програмування дозволяє з успіхом розв'язувати багато задач з економіки. Розглянемо одну з них. В нашому розпорядженні є якийсь запас коштів (ресурсів) K , який повинен бути розподілений між m підприємствами $\Pi_1, \Pi_2, \dots, \Pi_m$. Кожне з підприємств Π_i при вкладенні в нього якихось коштів x приносить дохід, який залежить від x , тобто представляє собою якусь функцію $\varphi_i(x)$. Всі функції $\varphi_i(x)$ ($i = 1, 2, \dots, m$) задані (функція не спадає). Завдання полягає в тому, щоб розподілити ресурси K між підприємствами так щоб в сумі вони дали максимальний дохід. Вирішуємо методом динамічного програмування. Хоча в своїй постановці воно не містить згадки про час, можна все ж операцію розподілу коштів подумки розгорнути в послідовності, вважаючи за перший крок вкладення коштів в підприємство Π_1 , за другий — в Π_2 и т.д.

Керована система S в даному випадку - кошти або ресурси, які розподіляються. Стан системи S перед кожним кроком характеризується одним числом S - готівковим запасом ще не вкладених коштів. У цьому завданні «кроковими управліннями» є кошти $x_1, x_2, \dots, x_m$ виділяються підприємствам. Потрібно знайти оптимальне управління, тобто таку сукупність чисел $x_1, x_2, \dots, x_m$ при якій сумарний дохід максимальний:

$$W = \sum_{i=1}^m \varphi_i(x_i) \Rightarrow \max$$

Знайдемо для кожного i -го кроку умовний оптимальний виграш (від цього кроку i до кінця), якщо ми підійшли до цього кроку з запасом коштів S . Позначимо умовний оптимальний виграш $W_i(S)$, а відповідне йому умовне оптимальне керування - кошти, вкладені в i -е підприємство, — $x_i(S)$.

Початок оптимізації виконуємо з останнього m -го кроку. Припустимо, що ми підійшли до цього кроку з залишком коштів S . Потрібно вкласти всю суму S цілком в підприємство $П_m$. Тому умовне оптимальне керування на m -м кроці: віддати останньому підприємству всі наявні засоби S , тобто:

$$x_m(S) = S$$

Умовний оптимальний виграш:

$$W_m(S) = \varphi_m(S)$$

Переймаючись цілою гамою значень S (розташовуючи їх досить тісно), ми для кожного значення S знатимемо $x_m(S)$ и $W_m(S)$. Останній крок оптимізований.

Перейдемо до передостаннього, $(m - 1)$ -му кроку. Нехай ми підійшли до нього з запасом засобів S . Позначимо $W_{m-1}(S)$ умовний оптимальний виграш на двох останніх кроках: $(m - 1)$ -м і m -м (який вже оптимізований). Якщо ми виділимо на $(m - 1)$ -м кроці $(m - 1)$ -му підприємству кошти x , то на останній крок залишиться $S - x$. Наш виграш на двох останніх кроках буде дорівнювати:

$$\varphi_{m-1}(x) + W_m(S - x)$$

Потрібно знайти таке x , при якому цей виграш максимальний:

$$W_{m-1}(S) = \max_{x \leq S} \{ \varphi_{m-1}(x) + W_m(S - x) \}$$

Знак $\max_{x < S}$ означає, що береться максимальне значення за всіма x , які тільки можливі (вкласти більше, ніж S , ми не можемо), від виразу, що стоїть у фігурних дужках. Цей максимум і є умовний оптимальний виграш за два останні кроки, а то значення x , при якому цей максимум досягається, - умовне оптимальне керування на $(m - 1)$ -м кроці. Далі мають бути оптимізовані $(m - 2)$ -й, $(m - 3)$ -й і т. д. кроки. Загалом, для будь-якого i -го кроку потрібно знаходити умовний оптимальний виграш за всі кроки, і до кінця по формулі

$$W_i(S) = \max_{x < S} \{ \varphi_i(x) + W_{i+1}(S - x) \}$$

і відповідне йому умовне оптимальне управління $x_i(S)$ - то значення x , при якому цей максимум досягається. Продовжуючи таким чином, дійдемо, нарешті, до 1-го підприємства *III*. Тут не потрібно буде варіювати значення S ; ми точно знаємо, що запас коштів перед першим кроком дорівнює K :

$$W^* = W_1(K) = \max_{x \leq K} \{ \varphi_1(x) + W_2(K - x) \}$$

Отже, максимальний дохід від усіх підприємств знайдений. Це значення x , при якому досягається максимум, і є оптимальне управління x_1^* на 1-м кроці. Після того як ми вкладемо ці кошти в 1-е підприємство, у нас їх залишиться $K - x_1^*$. Виділяємо другому підприємству оптимальну кількість коштів: $x_2^* = x_2(K - x_1^*)$, і т.д. до кінця [7].

В статті [8] методами комп'ютерного моделювання досліджено задачу оптимального розподілу інвестицій в умовах статистичної невизначеності. Було розроблено програмне забезпечення для розв'язання задачі розподілу інвестицій в таких умовах. Запропоновано різні варіанти критеріїв оптимальності. Як приклад, розглянуто задачу з вхідними даними, що підпорядковуються закону неперервного рівномірного розподілу з 10% відхиленнями від детермінованих значень. Показано, що у цьому випадку можливі декілька варіантів планів, які

для конкретних реалізацій випадкових параметрів можуть бути оптимальними. Наведено аналіз розв'язків за п'ятьма критеріями оптимальності.

У статті [9] сформульована задача розподілу коштів на різні заходи з управління ризиками всередині підприємства, що включає різні підрозділи. Обґрунтовано вибір моделі динамічного програмування для вирішення даного завдання, сформульовані основні залежності, функціональні рівняння, функції переходу стану системи.

У статті [10] наводиться доказ принципу оптимальності на основі його геометричного уявлення, формулюється завдання динамічного програмування в загальному вигляді і пропонується її рекурентне співвідношення. Основна суть динамічного програмування розглянута на прикладі підприємства, що займається розведенням і продажем риби. Функціональна залежність доходу підприємства від кількості продаваної риби характеризує стан процесу, тобто параметри стану. За допомогою математичних операцій виводиться рекуррентная формула, що дозволяє обчислити оптимальний дохід на будь-якому етапі. З урахуванням практичної значущості запропонована модель ускладнюється і надалі дохід підприємства розглядається в залежності від різного виду продукції, що виробляється.

У статті [11] пропонуються алгоритми вирішення стохастичних задач динамічного програмування великих розмірностей. Розглядається задача розподілу ресурсів.

1.3 Надійність електронного пристрою

Надійність – властивість об'єкта зберігати здатність до виконання своїх функцій у заданих умовах експлуатації в період певного часу. Під об'єктом тут мається на увазі елемент розрахунку надійності (резистор, конденсатор, транзистор, діод, мікросхема, вузол, пристрій, прилад або система), що має свій кіль-

кісний показник надійності. Радіоелектронне і електронне обладнання сучасних космічних апаратів включає в себе до десятка мільйонів складних пристроїв, тому проблема підвищення надійності має вкрай важливе значення [12].

На надійність апаратури великий вплив мають кліматичні умови, наприклад, температура, тиск, вологість, чистота повітря і т.д. В експлуатаційних умовах можуть виникнути впливи механічного характеру – удари, лінійні прискорення та вібрації, а також має великий вплив обслуговуючого персоналу при відвантаженні, транспортуванні, упакованні тощо.

Розрахувати надійність – значить визначити її кількісні характеристики за відомими характеристиками елементів, з яких складається апаратура. При цьому необов'язково обчислювати всі кількісні характеристики надійності. Більшість характеристик надійності взаємопов'язані між собою, тому для оцінки надійності достатньо буде обчислити обмежене число характеристик.

Розраховуючи надійність, метод буде залежити від способу з'єднання елементів. Розрізняють два з'єднання елементів – основне та резервне. Основне з'єднання елементів – це з'єднання, при якому відмова будь-якого елемента веде до відмови всього з'єднання. При цьому елементи можуть мати послідовне, паралельне, або змішане електричне з'єднання. Резервне з'єднання елементів – це з'єднання, при якому відмова з'єднання настає тільки в випадку, коли відмовили основний і всі резервні елементи.

Розглянемо рекомендований порядок повного розрахунку надійності:

1) кожен елемент пристрою розділяється на кілька груп залежно від типу елементів, розраховується кількість груп k і кількість елементів у кожній групі n_i ;

2) окремо для кожного елемента потрібно розрахувати коефіцієнт навантаження, також визначаються коефіцієнт експлуатації, робоча температура навколишнього середовища й інші фактори, які пов'язані з застосуванням даного елемента;

3) за спеціальними таблицями визначають значення інтенсивності відмов (подія, що полягає в порушенні працездатності виробу) для кожного ти-

пу елементів λ_i . Якщо для даного типу елементів немає значення λ_i , то береться середньо-групове значення для даної групи елементів;

4) обчислюються значення експлуатаційних інтенсивностей відмов для елементів даного типу елементів за формулою

$$\lambda_{\Sigma} = \lambda_i K_{\Sigma} \cdot \prod_{i=1}^m K$$

де m - число поправочних коефіцієнтів для даної групи елементів;

5) обчислюються добутки $\pi_{\alpha} \cdot \lambda$, що характеризують частку відмов, внесених елементами даної групи в загальну інтенсивність відмов пристрою;

6) обчислюється загальна інтенсивність відмов пристрою з урахуванням режимів роботи й умов застосування елементів шляхом додавання добутків $\pi_{\alpha} \cdot \lambda_{\Sigma i}$ по всім k групах за формулою

$$\lambda = \sum_{i=1}^k \lambda_{\Sigma i} \cdot n_i$$

7) розраховується ймовірність безвідмовної роботи пристрою за формулою $P(t) = e^{-\lambda t}$ для заданого часу безперервної справної роботи t_0 ;

8) визначається напрацювання на відмову за формулою $T_0 = 1 / \lambda$.

Коли потрібно розрахувати надійність складного пристрою, то за зручності роблять його поділ на окремі плати, каскади або вузли. Розраховують інтенсивність відмов з початку окремих плат, каскадів або вузлів, а потім визначають інтенсивність відмов усього пристрою шляхом додавання інтенсивностей відмов плат, каскадів або вузлів, що входять до нього. За результатами розрахунку будуються залежність $P(t)$ та отримані дані порівнюються із заданими значеннями в ТУ й робиться висновок: чи відповідає спроектований пристрій необхідній надійності [13].

1.4 Загальні методи розрахунку надійності

В навчальному посібнику [14] розглянуто фундаментальні основи теорії надійності, методи розрахунку показників експлуатаційної надійності, метрологічної надійності та надійності програмного забезпечення, а також методи підвищення надійності технічних засобів. Для опису надійності розглянемо наступні методи:

- метод інтегральних рівнянь;
- метод диференціальних рівнянь;
- метод оцінки надійності за графом можливих станів пристрою.

Метод інтегральних рівнянь можна застосовувати при розрахунку надійності будь - яких приладів при будь - яких розподілах часу безвідмовної роботи і часу відновлення.

В данному методі, визначення показників надійності відбувається шляхом складання і вирішення інтегральних або інтегрально - диференціальних рівнянь. При складанні інтегральних рівнянь виділяють нескінченно малі інтервали часу. Для цих інтервалів часу розглядають складні події, котрі з'являються при спільній дії декількох факторів.

Дані рівняння порівняно просто складати, але важко розв'язати. Часто розв'язок доводиться знаходити чисельними методами за допомогою ЕОМ. У зв'язку з цим метод інтегральних рівнянь у даний час не одержав широкого розповсюдження.

Метод диференціальних рівнянь був створений на припущенні того, що час між відмовами і час відновлення підлягають показниковим розподілам. При цьому параметр потоку відмов $\Omega = \lambda = 1/T_{\text{ср}}$, а інтенсивність відмови $\mu = 1/T_{\text{в}}$, де $T_{\text{ср}}$, $T_{\text{в}}$ – відповідно середній час до відмови і час відновлення.

Для визначення показників надійності складають і розв'язують систему диференціальних рівнянь для ймовірностей станів (рівнянь Колмогорова). Щоб

при цьому гранично зменшити витрати праці на розрахунок зазвичай припускають, що:

- об'єкти, що відмовили, починають негайно відновлювати;
- відсутні обмеження на кількість відновлень;
- надійність засобів контролю ідеальна.

Математичну модель зображують у вигляді графа станів. На цьому графі можливі стани приладу при відмовах його елементів зображують кружочками. Можливі напрямки переходів з одного стану в інший – стрілками. Біля стрілок вказують інтенсивність переходів (наприклад, λ і μ).

Зобразимо приклад такого графа на рисунку 1.1

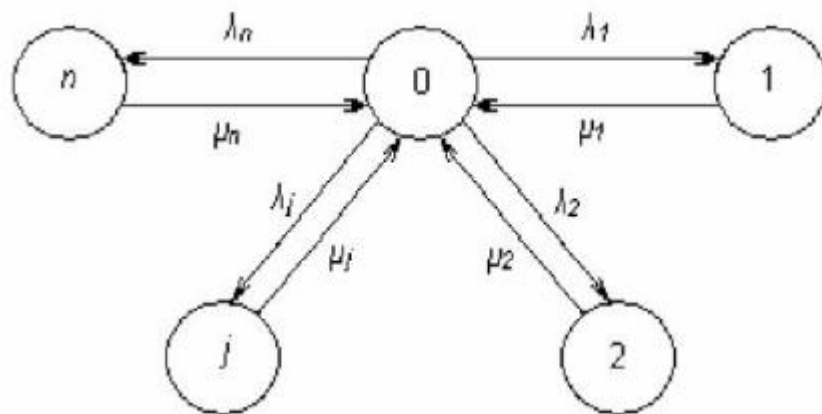


Рисунок 1.1 – граф станів

В ліву частину кожного рівняння ставлять похідну за часом від ймовірності перебування пристрою в j -му стані в момент часу t . Кількість членів у правій частині дорівнює кількості стрілок, що з'єднують розглянутий стан з іншим. При цьому кожен член дорівнює ймовірності переходу з одного стану в інший, а саме добуткові інтенсивності переходу на ймовірність того i -го стану, з якого стрілка виходить. Отримана система диференціальних рівнянь доповнюється нормованою умовою

$$\sum_{j=0}^{n+1} P_j(t) = 1$$

де $P_j(t)$ – ймовірність перебування пристрою в j -му стані;

$n + 1$ – кількість можливих станів.

Далі вся множина станів розбивається на дві підмножини:

а) n_1 – підмножина станів, у якому прилад нероботоздатний;

б) n_2 – підмножина станів, у яких прилад роботоздатний.

Тоді функцію готовності можна визначити як:

$$G(t) = \sum_{j=0}^n P_j(t)$$

де $P_j(t)$ – ймовірність перебування пристрою в j -му роботоздатному стані.

Якщо необхідно визначити коефіцієнт готовності (або простою) розглядають сталий режим експлуатації при $t \rightarrow \infty$.

У цьому випадку всі похідні $P'_j(t) = 0$ і система диференціальних рівнянь переходить у систему алгебраїчних рівнянь.

Метод оцінки надійності за графом можливих станів заснований на методі диференціальних рівнянь, при якому доводиться розв'язувати систему лінійних алгебраїчних рівнянь. Структура визначників цієї системи дозволяє сформулювати правило перебування виразів безпосередньо за графом. Таке правило для виразів стаціонарної ймовірності перебування в j -му стані полягає в такому: проходять найкоротші шляхи (без повернення) із усіх крайніх станів у кожен стан за напрямком стрілок і перемножують усі інтенсивності переходів. Кожна інтенсивність переходу враховується тільки один раз. Ймовірність перебування в j -му стані для графів без кілець визначається за формулою:

$$P_j(t) = \frac{\Delta_j}{\sum_{i=0}^{K+1} \Delta_i},$$

де Δ_i, Δ_j – добуток інтенсивностей переходів із усіх найкоротших станів відповідно в j -ті та i -ті при русі за найкоротшим шляхом в напрямку стрілок, $(K + 1)$ – кількість станів.

Стани, котрі вважають найкоротшими, ті які не мають вихідних стрілок при невідновлюваному і мають не більше однієї вихідної стрілки при відновлюваному пристрою. Застосовуючи дане правило можна отримати формулу для коефіцієнта готовності без складання і розв’язання диференціальних рівнянь.

2 ЗАДАЧА ПРО НАДІЙНІСТЬ ЕЛЕКТРОННОГО ПРИСТРОЮ

2.1 Постановка задачі

Розглянемо таку задачу про надійність електронного пристрою. Конструюється електронний прилад, що складається з j основних компонентів. Всі компоненти з'єднані послідовно, тому вихід з ладу однієї з них призводить до відмови всього приладу. Надійність приладу можна підвищити шляхом дублювання кожної компоненти. Кожна компонента може складатися з x_i блоків. Загальна вартість приладу не повинна перевищувати y грошових одиниць. Кожна компонента j , що складається з x_i блоків, характеризується вартістю C_{ij} та надійністю R_{ij} . Потрібно визначити кількість блоків x_j в компоненті j , щоб надійність приладу була максимальною, а ціна не перевищувала заданої величини.

Але на практиці надійність компонентів приладу може змінюватися в деякому діапазоні. Тому актуальним є розв'язання задачі в умовах невизначеності.

Як приклад розглянемо задачу про надійність електронного пристрою з наступними вхідними даними: $y = 10, n = 3, m = 3$, C_{ij} наведені в таблиці; R_{ij} підпорядковуються закону нормального розподілу з 10%, 20% та 30% відхиленнями від значень, наведених в таблиці 2.1.

Таблиця 2.1 – Вхідні дані

x_i	$j = 1$		$j = 2$		$j = 3$	
	C_1	R_1	C_2	R_2	C_3	R_3
1	1	0.6	3	0.7	2	0.5
2	2	0.8	5	0.8	4	0.7
3	3	0.9	6	0.9	5	0.9

2.2 Програмна реалізація задачі про надійність з нормально розподіленими параметрами

Будемо розв'язувати задачу методом перебору, який реалізуємо за допомогою пакету MATLAB [15-16].

Вхідними даними програми є :

- y - загальна вартість приладу,
- n - кількість компонентів в приладі,
- m - максимальна кількість блоків в компоненті,
- C - вартість,
- R - надійність.

Вхідну матрицю R_{ij} , що підпорядковується нормальному закону розподілу з 10%, 20%, 30% відхиленням від значень, будемо задавати за допомогою функції `normrnd(MU,SIGMA)` пакету Statistics Toolbox програмного середовища MATLAB. Функція `normrnd(MU, SIGMA)` – призначена для генерації псевдовипадкових чисел за нормальним законом для кожної пари параметрів: MU – математичне очікування, $SIGMA$ – середньоквадратичне відхилення. Для кожного випадку відхилення нам потрібно підібрати $SIGMA$.

Матриці для нормального закону потрібно виписати. Для цього необхідно визначити сигма для кожного числа в умови матриці R .

Розглянемо для прикладу перший елемент матриці R для 10% відхилення по нормальному закону. Спочатку, цей елемент дорівнює 0,6. Для нього потрібно визначити сигма для 10% відхилення. 10% відхилення від числа 0,6 означає, що значення має змінюватися в наступному діапазоні: від $0,6 * 0,9$ до $0,6 * 1,1$, тобто, в діапазоні від 0,54 до 0,66 . І таким же чином підбираємо значення матриці для 20% відхилення і 30% відхилення за допомогою MATLAB.

```

>> R = normrnd (0.7,0.0120,1000,1000);
>> max(max(R))

ans =

    0.7600

>> min(min(R))

ans =

    0.6392
fx >> |

```

Рисунок 2.1 – Вирізка прикладу розрахунку значень SIGMA

Провівши розрахунки в MATLAB, було отримано наступні матриці для кожного варіанта відхилення:

- Матриця R для 10% відхилення матиме такий вид:

$R = [\text{normrnd}(0.6,0.0125) \text{ normrnd}(0.7,0.0120) \text{ normrnd}(0.5,0.0120)$
 $\text{normrnd}(0.8,0.0179) \text{ normrnd}(0.8,0.0179) \text{ normrnd}(0.7,0.0120)$
 $\text{normrnd}(0.9,0.0190) \text{ normrnd}(0.9,0.0190) \text{ normrnd}(0.9,0.0190)] ;$

- Матриця R для 20% відхилення матиме такий вид:

$R = [\text{normrnd}(0.6,0.026) \text{ normrnd}(0.7,0.028) \text{ normrnd}(0.5,0.020)$
 $\text{normrnd}(0.8,0.034) \text{ normrnd}(0.8,0.034) \text{ normrnd}(0.7,0.028)$
 $\text{normrnd}(0.9,0.036) \text{ normrnd}(0.9,0.036) \text{ normrnd}(0.9,0.036)];$

- Матриця R для 30% відхилення матиме такий вид:

$R = [\text{normrnd}(0.6,0.039) \text{ normrnd}(0.7,0.044) \text{ normrnd}(0.5,0.029)$
 $\text{normrnd}(0.8,0.05) \text{ normrnd}(0.8,0.05) \text{ normrnd}(0.7,0.044)$
 $\text{normrnd}(0.9,0.055) \text{ normrnd}(0.9,0.055) \text{ normrnd}(0.9,0.055)];$

Далі було створено програмний код за допомогою мови MATLAB, який розраховує всі оптимальні плани та значення надійності, і заносить дані в Excel. (див. додаток А).

В програмі реалізовано метод перебору знаходження оптимального розв'язку задачі. Спочатку ми створюємо масив розмірністю 1×1000 .

Беремо до уваги, що x_1 – описує кількість блоків в кожній компоненті, за умовою задачі, компонента може містити від одного до трьох блоків, відповідно x_1 може приймати значення від 1 до 3. Аналогічно x_2 и x_3 також можуть приймати значення від 1 до 3. Формуємо вектор всіх можливих наборів.

Далі в програмі реалізуємо умови, які перебирають всі можливі плани, відсортовують їх та автоматично заносять до файлу Excel, який потім зберігаємо в форматі CSV для зручного використання даних в подальшому.

В результаті ми отримали 1000 запусків програми для кожного з варіантів відхилень. В таблиці Excel було зафіксовано оптимальні плани та значення надійності (Рисунок 2.2-2.4).

	A	B
1	213	312
2	0,534765	0,450522
3	0,499061	
4	0,493458	
5	0,497024	
6	0,514965	
7	0,508833	
8	0,484575	
9	0,48823	
10	0,506684	
11	0,523995	
12	0,50738	
13	0,506033	
14	0,485423	
15	0,494631	
16	0,503304	
17	0,509718	
18	0,513316	
19	0,500262	
20	0,479366	
21	0,50147	
22	0,5065	
23	0,530504	
24	0,480716	
25	0,496114	
26	0,480725	
27	0,512864	
28	0,510934	
29	0,490289	
30	0,514405	
31	0,495033	

	A
969	0,525104
970	0,478345
971	0,515562
972	0,501837
973	0,512652
974	0,515073
975	0,508427
976	0,495595
977	0,543239
978	0,486329
979	0,493214
980	0,492024
981	0,487937
982	0,511531
983	0,483043
984	0,521696
985	0,489821
986	0,507945
987	0,509093
988	0,487646
989	0,492112
990	0,530446
991	0,463317
992	0,521186
993	0,509028
994	0,495712
995	0,490959
996	0,549144
997	0,486744
998	0,486346
999	0,474056

Рисунок 2.2 – Вирізка з таблиці значень вибірки для 10%

	A	B	C		
1	213	312	321	910	0,539803
2	0,530855	0,487359	0,410507	911	0,52594
3	0,493074	0,477149		912	0,461216
4	0,527145	0,493594		913	0,512596
5	0,493107	0,492409		914	0,482946
6	0,502199	0,51178		915	0,567826
7	0,491214	0,481281		916	0,600309
8	0,494188	0,459992		917	0,461764
9	0,513833	0,51449		918	0,514685
10	0,434774	0,425398		919	0,533803
11	0,538839	0,467372		920	0,498567
12	0,488648	0,455168		921	0,563633
13	0,49137	0,510017		922	0,516865
14	0,512417	0,520798		923	0,486619
15	0,50281	0,556204		924	0,502219
16	0,514265	0,519101		925	0,507134
17	0,489699	0,462843		926	0,439131
18	0,475457	0,482907		927	0,49626
19	0,478648	0,510967		928	0,545086
20	0,531462	0,480453		929	0,506401
21	0,554916	0,466304		930	0,542196
22	0,489946	0,452625		931	0,502142
23	0,545753	0,502323		932	0,527807
24	0,563749	0,431287		933	0,459833
25	0,499646	0,463247		934	0,496919
26	0,506215	0,495755		935	0,506759
27	0,534772	0,441943		936	0,521627
28	0,539958	0,476078		937	0,552098
29	0,540131	0,415558		938	0,541463
30	0,575285	0,494213		939	0,594461
31	0,568965	0,479483		940	0,543566

Рисунок 2.3 – Вирізка з таблиці значень вибірки для 20%

	A	B	C	D	E
1	122	213	231	312	321
2	0,407693	0,484468	0,455686	0,462312	0,437021
3		0,441108	0,402636	0,561744	0,41329
4		0,581612		0,455441	0,435704
5		0,528924		0,450688	0,417526
6		0,513418		0,475643	
7		0,57208		0,441115	
8		0,526682		0,469544	
9		0,573289		0,485137	
10		0,488708		0,451898	
11		0,50452		0,464074	
12		0,483704		0,519486	
13		0,50039		0,466181	
14		0,406438		0,492599	
15		0,430032		0,477938	
16		0,502344		0,524393	
17		0,51591		0,459853	
18		0,463261		0,502782	
19		0,4915		0,403663	
20		0,496678		0,493713	
21		0,453252		0,485048	
22		0,509917		0,48573	
23		0,435186		0,442901	
24		0,502505		0,425526	
25		0,516183		0,533169	
26		0,503634		0,425252	
27		0,487399		0,535714	
28		0,446796		0,437422	
29		0,499771		0,493896	
30		0,617233		0,516738	
31		0,494643		0,509104	

Рисунок 2.4 – Вирізка з таблиці значень вибірки для 30%

2.3 Результати розрахунків задачі про надійність з нормально розподіленими параметрами

Для кожного з варіантів відхилення було зроблено 1000 запусків програмного коду. Результати зображенні на діаграмах “Ящик з вусами”, що були побудовані за допомогою мови R (Рисунок 2.4 – 2.6). По горизонталі вказано кількість блоків для кожної компоненти. По вертикалі величину надійності прибо­ра. По довжині ящика зображено міжквартильний розмах вибірки для кожного із оптимальних наборів. Середня лінія, зображує середнє значення вибірки. Вуси показують повний розмах даних. На кінцях вусів зображено максимальне і мінімальне значення вибірки.

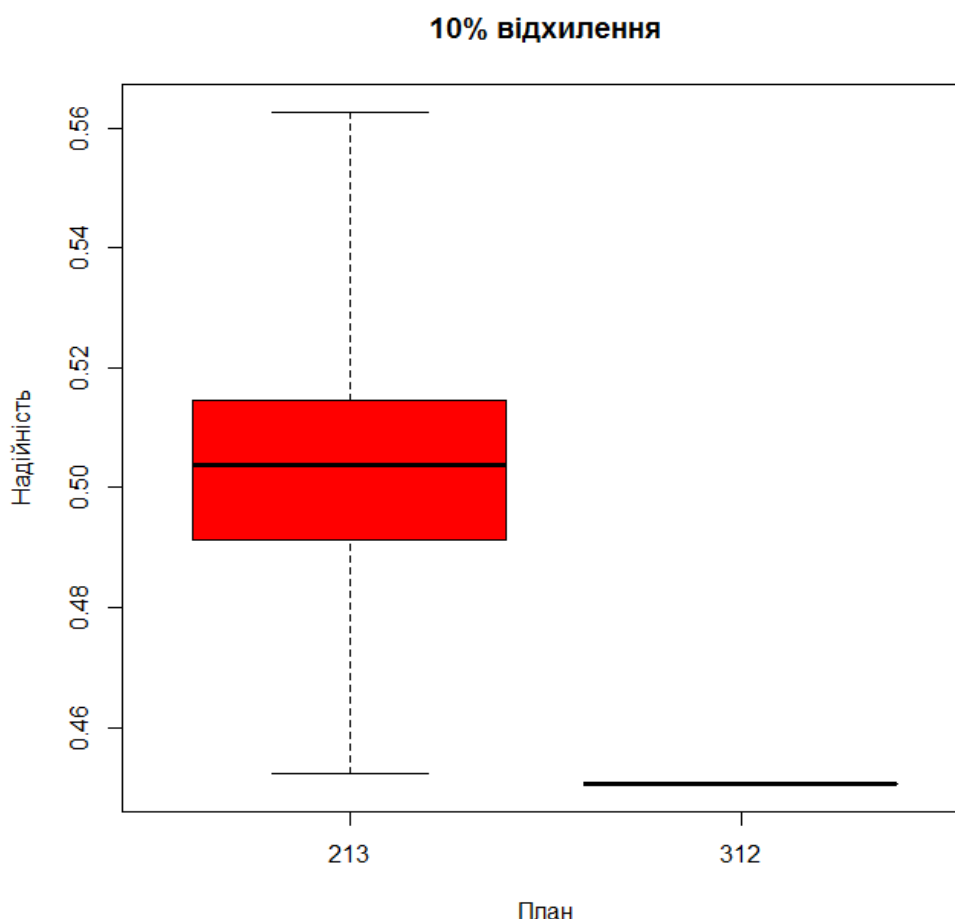


Рисунок 2.5 – “Ящик з вусами” для 10% відхилення

Для 10% відхилення отримали два набори оптимальних планів; (2,1,3) та (3,1,2). Перший має значну перевагу.

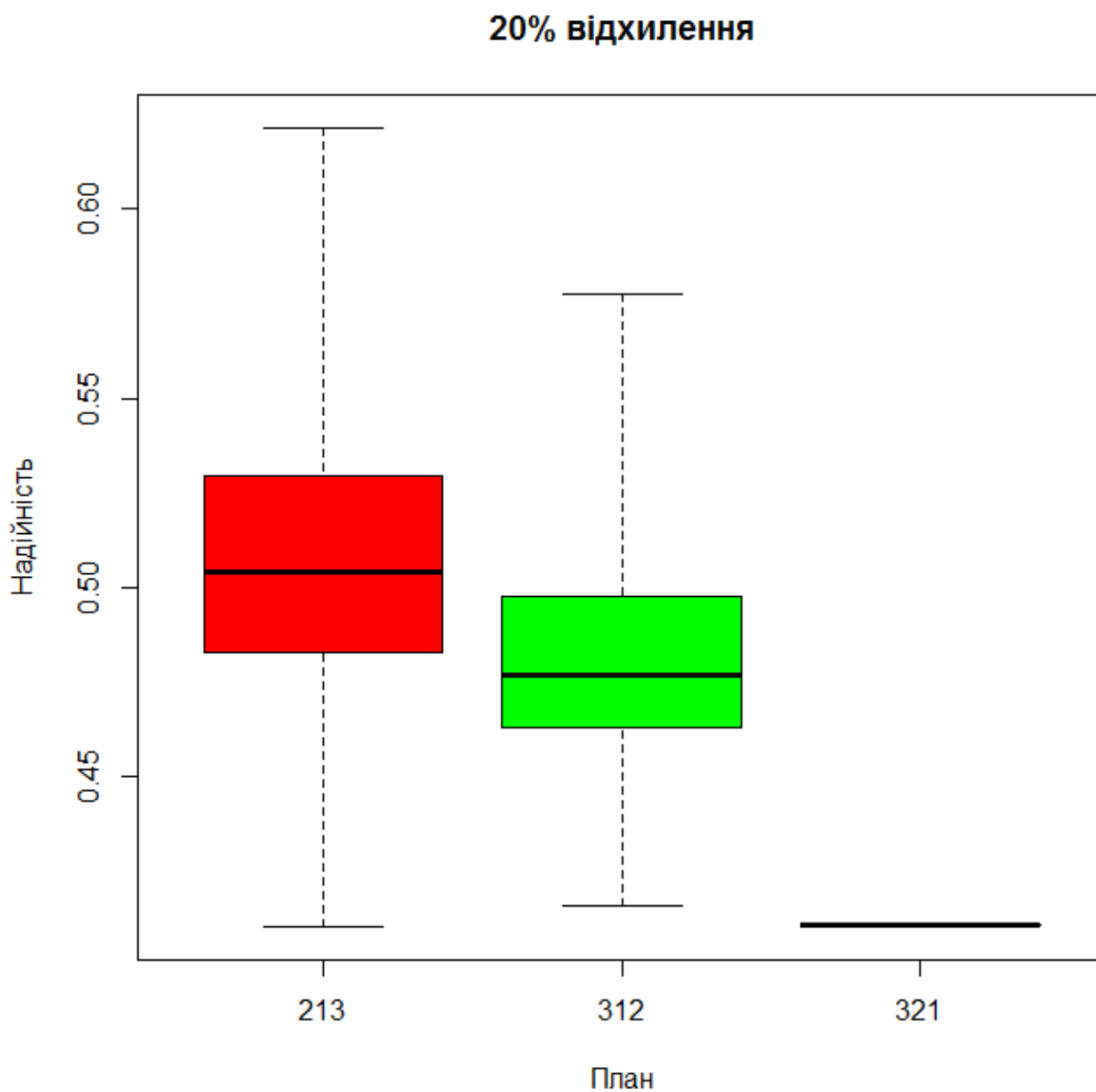


Рисунок 2.6 – “Ящик з вусами” для 20% відхилення

Для 20% відхилення отримали три набори оптимальних планів; (2,1,3), (3,1,2) та (3,2,1).

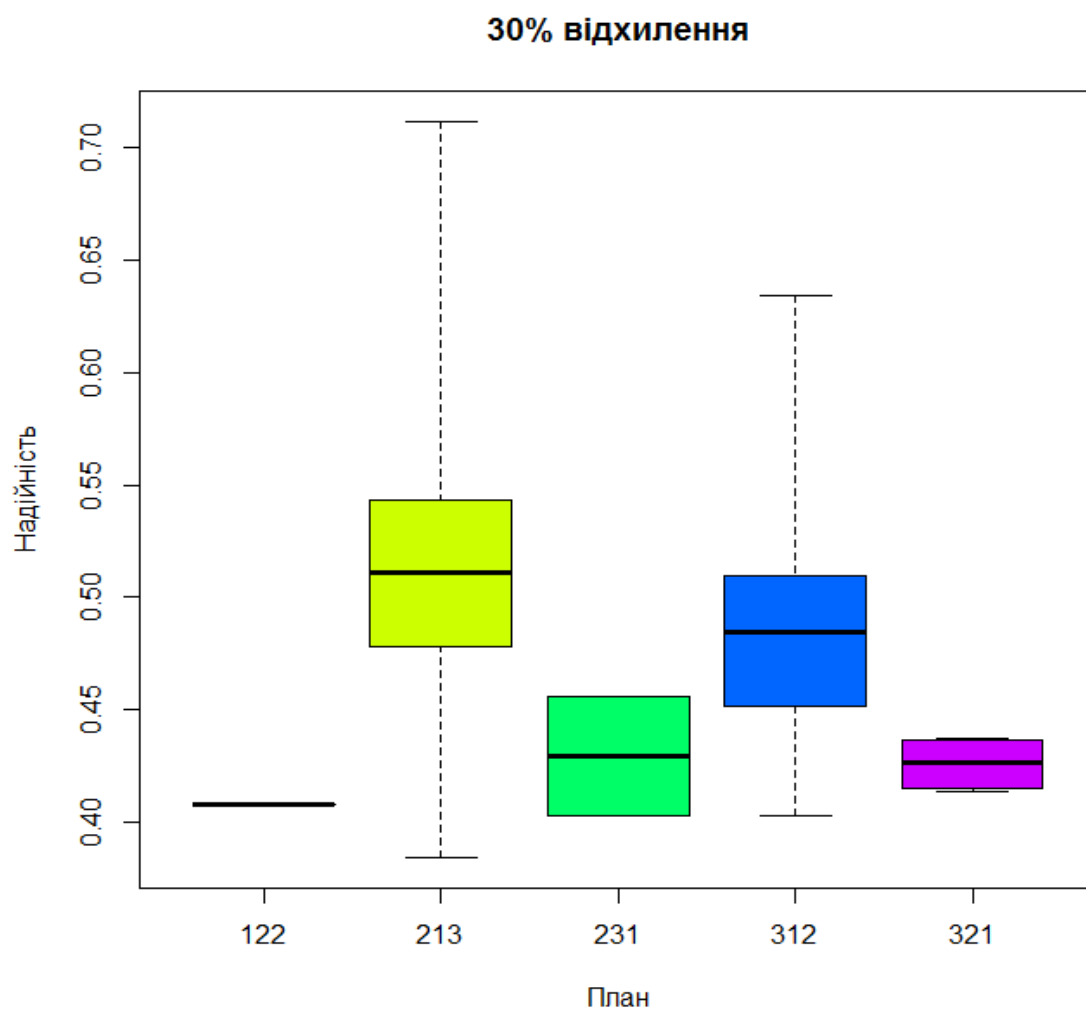


Рисунок 2.7 – “Ящик з вусами” для 30% відхилення

Для 30% відхилення отримали п’ять наборів оптимальних планів; (1,2,2), (2,1,3), (2,3,1) (3,1,2) та (3,2,1).

2.4 Аналіз розрахунків задачі задачі про надійність з нормально розподіленими параметрами

Необхідно обрати один із варіантів, що забезпечить максимальну надійність. Проаналізуємо отримані результати за такими критеріями оптимальності:

- кількість відсотків появи кожного з плану;
- середньоарифметична надійність;
- ймовірність того, що буде отримано результат вище ніж 0,45;
- ймовірність того, що буде отримано результат вище ніж 0,5;
- ймовірність того, що буде отримано результат вище ніж 0,55.

Дані розрахунку за цими критеріями зображені у таблицях (2.2-2.4).

Таблиця 2.2 – Результати за обраними критеріями для 10% оптимальності

	213	312
Кількість % появи кожного з планів	99,9%	0,1%
Середньоарифметична надійність	0,5036058	0,4505225
> 0,45	100%	100%
> 0,5	99,5%	0%
> 0,55	0,3%	0%

Таблиця 2.3 – Результати за обраними критеріями для 20% оптимальності

	213	312	321
Кількість % появи кожного з варіантів	94,1%	5,8%	0,1%
Середньоарифметична надійність	0,5061108	0,480289	0,410507
> 0,45	94,8%	84,4%	0%
> 0,5	55,4%	24,1%	0%
> 0,55	11,6%	3,4%	0%

Таблиця 2.3 – Результати за обраними критеріями для 30% оптимальності

	122	213	231	312	321
Кількість % поява кожного з варіантів	0,1%	86%	0,2%	13,3%	0,4%
Середньоарифметична надійність	0,4076935	0,511501	0,4291613	0,483437	0,425886
> 0,45	0%	89,7%	50%	78,1%	0%
> 0,5	0%	58,3%	0%	31,5%	0%
> 0,55	0%	20,1%	0%	7,51%	0%

Проаналізувавши розглянуті критерії для кожного варіанту оптимальності, можна зробити висновок, що план (2 1 3) дає найвищу надійність.

Для розглянутого прикладу задачі про надійність з нормально розподіленими вхідними параметрами, для кожного варіанту відхилення – 10%, 20% та 30% оптимальним розв’язком є план (2 1 3). Тобто, прибор містить 3 компоненти: перша повинна складатися з 2 блоків, друга – з 1 блока та третя – з 3 блоків.

Дану задачу було розв’язано методом перебору і визначено оптимальний розв’язок, який не перевищує бюджет і дає максимальну надійність.

ВИСНОВОК

У дипломній роботі було зроблено наступне:

- розроблено програмний код для розв'язання задачі про надійність з нормально розподіленими вхідними параметрами методом перебору даних, що розраховує оптимальний план та значення надійності, і заносить дані в Excel;
- отримано розв'язок задачі з нормально розподіленими параметрами з трьома варіантами відхилень;
- запропоновано різні варіанти критеріїв оптимальності;
- проаналізовано розв'язок задачі за п'ятьма критеріями оптимальності.

ПЕРЕЛІК ПОСИЛАНЬ

1. **Беллман, Р.** Динамическое программирование [Текст] / Р. Беллман. – М. : ИЛ, 1960. – 400 с.
2. **Лежнёв, А.В.** Динамического программирование в экономических задачах: учебное пособие [Текст] / А.В. Лежнёв. – М. : БИНОМ. Лаборатория знаний. – 2010. – 176 с.
3. **Таха, Х. А.** Введение в исследование операций / Х. А. Таха ; пер. с англ. – 7-е изд. – М. : "Вильямс". – 2005. – 912 с.
4. **Канцедал, С. А.** . Конструирование и исследование алгоритмов решения задачи про рюкзак [Текст] / С. А. Канцедал // Автомобильный транспорт. – 2015. – № 36. – С. 154 – 160.
5. **Куприяшин, М.А.** Исследование алгоритма точного решения задачи о рюкзаке методом динамического программирования [Текст] / М.А. Куприяшин, Г.И. Борзунов // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. – 2016. - № 17. – С. 121 – 130.
6. **Кагиров, Р. Р.** Многомерная задача о рюкзаке: новые методы решения [Текст] / Р. Р. Кагиров // Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. Математика, механика, информатика. – 2015. – № 5. – С. 16 – 20.
7. **Кірік, О. Є.** Розподіл ресурсів у розподільчих системах з оптимальним перерозподілом навантаження постачальників продукту / О. Є. Кірік // Системні дослідження та інформаційні технології. – 2013. – № 4. – С. 38–51.
8. **Подковаліхіна, О.О.** Задача розподілу інвестицій в умовах статистичної невизначеності / О.О. Подковаліхіна, В.О. Логвіненко, В.Є. Бахрушин // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 2 (121). – Дніпро, 2-19. – С. 56-63.

9. **Ростова, Е.П** Постановка задачи динамического программирования для распределения средств по управлению рисками на предприятии [Текст] / Е. П Ростова // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2013. — №6. — С. 152–155.
10. **Сутягина, Н. И.** Метод динамического программирования при принятии микроэкономического решения [Текст] / Н. И. Сутягина // Вестник НГИЭИ. — 2014. – № 11 (42). — С. 72–77.
11. **Докучаев, А. В.** Алгоритмы решения стохастических задач динамического программирования большой размерности [Текст] / А. В. Докучаев, А. П. Котенко // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. Науки. – 2008. – № 2 (17). — С. 203–209.
12. **Половко А. М.** Основы теории надёжности / А. М. Половко // — М.:Наука. — 1964. — 446с.
13. **Маликов И.** Основы теории и расчёта надёжности / И. М. Маликов, А. М. Половко, П. А.// — Л.:Судпромгиз. – 1959. – 325 с.
14. **Васілевський О. М.** Нормування показників надійності технічних засобів: навчальний посібник/ О. М. Васілевський, В. О. Поджаренко// — Вінниця: ВНТУ. — 2010. — 129 с.
15. **Гонсалес, Р.С.** Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс // Техносфера. – 2006. – 616 с.
16. **Дьяконов В. П.** Компьютерная математика. Теория и практика / В. П. Дьяконов// — СПб: «Питер». 2001. — 1296 с.

ДОДАТОК А

Код програмної реалізації задачі про надійність з нормально розподіленими
вхідними параметрами

```
mas111=zeros(1,1000);  
mas112=zeros(1,1000);  
mas113=zeros(1,1000);  
mas121=zeros(1,1000);  
mas122=zeros(1,1000);  
mas123=zeros(1,1000);  
mas131=zeros(1,1000);  
mas132=zeros(1,1000);  
mas133=zeros(1,1000);  
mas211=zeros(1,1000);  
mas212=zeros(1,1000);  
mas213=zeros(1,1000);  
mas221=zeros(1,1000);  
mas222=zeros(1,1000);  
mas223=zeros(1,1000);  
mas231=zeros(1,1000);  
mas232=zeros(1,1000);  
mas233=zeros(1,1000);  
mas311=zeros(1,1000);  
mas312=zeros(1,1000);  
mas313=zeros(1,1000);  
mas321=zeros(1,1000);  
mas322=zeros(1,1000);  
mas323=zeros(1,1000);  
mas331=zeros(1,1000);  
mas332=zeros(1,1000);  
mas333=zeros(1,1000);  
counter111=1;  
counter112=1;  
counter113=1;  
counter121=1;  
counter122=1;  
counter123=1;  
counter131=1;  
counter132=1;  
counter133=1;  
counter211=1;  
counter212=1;  
counter213=1;  
counter221=1;  
counter222=1;  
counter223=1;  
counter231=1;  
counter232=1;  
counter233=1;
```

```

counter311=1;
counter312=1;
counter313=1;
counter321=1;
counter322=1;
counter323=1;
counter331=1;
counter332=1;
counter333=1;
chek=0;
while chek<1000
y1 = 10 ;
C = [ 1 3 2
      2 5 4
      3 6 5 ] ;
R = [ normrnd(0.6,0.0125) normrnd(0.7,0.0120) normrnd(0.5,0.0120)
      normrnd(0.8,0.0179) normrnd(0.8,0.0179) normrnd(0.7,0.0120)
      normrnd(0.9,0.0190) normrnd(0.9,0.0190) normrnd(0.9,0.0190)] ;
[u9,n] = size(C);
p9 = y1 + 1;
minimum_y = zeros(1, n+1);
for i = n : -1 : 1
    minimum_y(i) = minimum_y(i +1) + C(1,i);
end
max_y = zeros(1, n);
max_y(1) = y1;
for i = 2 : n
    max_y(i) = max_y(i -1) - C(1, i -1);
end
max_x = sum(isfinite(R));
p8 = max(max_y - minimum_y(1:n) +1);
X = -inf * ones(p8, n +1);
L = X;
L(:, n +1) = ones(p8, 1);
for i = n : -1 :1
p7 = max_y(i) - minimum_y(i) + 1;
u7 = max_x(i);
Fsu = -inf * ones(p7,u7);
for p = 1 : p7
    for u = 1 : u7
        Yj = minimum_y(i) + p - 1;
        p1 = (Yj - C(u,i)) - minimum_y(i +1) + 1;
        if 1 <= p1 && p1 <= p7
            Fsu(p, u) = R(u, i) * L(p1, i+1);
        end
    end
end
end
[fs, ind] = max(Fsu');
L(1: p7, i) = fs';
X(1: p7, i) = ind';
%disp(['Таблица' int2str(i)])
%disp([ blanks(u7*3 -2) ])
%disp([(minimum_y(i) : max_y(i))', fs', ind', Fsu ])

```

```

end
nabor = zeros(1, n);
Yj = y1;
for i = 1 : n
    p = Yj - minimum_y(i) + 1;
    nabor(i) = X(p,i);
    Yj = Yj - C(nabor(i),i);
end
%disp('_____Відповідь_____')
nabor;
reliability = L(p8, 1);

resultat=nabor(1)*100+nabor(2)*10+nabor(3);

if resultat==111
    mas111(counter111)=reliability;
    counter111=counter111+1;
end
if resultat==121
    mas121(counter121)=reliability;
    counter121=counter121+1;
end
if resultat==122
    mas122(counter122)=reliability;
    counter122=counter122+1;
end
if resultat==123
    mas123(counter123)=reliability;
    counter123=counter123+1;
end
if resultat==131
    mas131(counter131)=reliability;
    counter131=counter131+1;
end
if resultat==132
    mas132(counter132)=reliability;
    counter132=counter132+1;
end
if resultat==133
    mas133(counter133)=reliability;
    counter133=counter133+1;
end
if resultat==211
    mas211(counter211)=reliability;
    counter211=counter211+1;
end
if resultat==212
    mas212(counter212)=reliability;
    counter212=counter212+1;
end
if resultat==213
    mas213(counter213)=reliability;
    counter213=counter213+1;

```

```

end
if resultat==221
    mas221(counter221)=reliability;
    counter221=counter221+1;
end
if resultat==222
    mas222(counter222)=reliability;
    counter222=counter222+1;
end
if resultat==223
    mas223(counter223)=reliability;
    counter223=counter223+1;
end
if resultat==231
    mas231(counter231)=reliability;
    counter231=counter231+1;
end
if resultat==232
    mas232(counter232)=reliability;
    counter232=counter232+1;
end
if resultat==233
    mas233(counter233)=reliability;
    counter233=counter233+1;
end
if resultat==311
    mas311(counter311)=reliability;
    counter311=counter311+1;
end
if resultat==312
    mas312(counter312)=reliability;
    counter312=counter312+1;
end
if resultat==313
    mas312(counter312)=reliability;
    counter312=counter312+1;
end
if resultat==321
    mas321(counter321)=reliability;
    counter321=counter321+1;
end
if resultat==322
    mas322(counter322)=reliability;
    counter322=counter322+1;
end
if resultat==323
    mas323(counter323)=reliability;
    counter323=counter323+1;
end
if resultat==331
    mas331(counter331)=reliability;
    counter331=counter331+1;
end

```

```

if resultat==332
    mas332(counter332)=reliability;
    counter332=counter332+1;
end
if resultat==333
    mas333(counter333)=reliability;
    counter333=counter333+1;
end
chek=chek+1;
end
xlswrite('10pro.csv',[111 112 113 121 122 123 131 132 133 211 212
213 221 222 223 231 232 233 311 312 313 321 322 323 331 332
333 ],'A1:AA1');
xlswrite('10pro.csv',mas111,'A2:A1001');
xlswrite('10pro.csv',mas112,'B2:B1001');
xlswrite('10pro.csv',mas113,'C2:C1001');
xlswrite('10pro.csv',mas121,'D2:D1001');
xlswrite('10pro.csv',mas122,'E2:E1001');
xlswrite('10pro.csv',mas123,'F2:F1001');
xlswrite('10pro.csv',mas131,'G2:G1001');
xlswrite('10pro.csv',mas132,'H2:H1001');
xlswrite('10pro.csv',mas133,'I2:I1001');
xlswrite('10pro.csv',mas211,'J2:J1001');
xlswrite('10pro.csv',mas211,'K2:K1001');
xlswrite('10pro.csv',mas213,'L2:L1001');
xlswrite('10pro.csv',mas221,'M2:M1001');
xlswrite('10pro.csv',mas222,'N2:N1001');
xlswrite('10pro.csv',mas223,'O2:O1001');
xlswrite('10pro.csv',mas231,'P2:P1001');
xlswrite('10pro.csv',mas232,'Q2:Q1001');
xlswrite('10pro.csv',mas233,'R2:R1001');
xlswrite('10pro.csv',mas311,'S2:S1001');
xlswrite('10pro.csv',mas312,'T2:T1001');
xlswrite('10pro.csv',mas313,'U2:U1001');
xlswrite('10pro.csv',mas321,'V2:V1001');
xlswrite('10pro.csv',mas322,'W2:W1001');
xlswrite('10pro.csv',mas323,'X2:X1001');
xlswrite('10pro.csv',mas331,'Y2:Y1001');
xlswrite('10pro.csv',mas332,'Z2:Z1001');
xlswrite('10pro.csv',mas333,'AA2:AA1001');

```