

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: **«ПРИЙНЯТТЯ РІШЕНЬ В СИСТЕМАХ КЕРУВАННЯ
ДЕКІЛЬКОМА ЗАПАСАМИ»**

Виконав: студент 4-го року навчання
освітньої програми «Прикладна
математика»,
спеціальності 113 Прикладна
математика

Малий Данило Костянтинівич

Керівник: Чорней Р. К.,
кандидат фіз.-мат. наук, доцент

Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____

«____» _____ 20____ р.

Календарний план виконання роботи:

Тема: Прийняття рішень в системах керування декількома запасами

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	02.11.2021	
2.	Огляд літератури за темою роботи	15.12.2021	
3.	Аналіз отриманих даних	15.01.2022	
4.	Написання текстової частини роботи	01.03.2022	
5.	Створення програмного застосунку	01.05.2022	
7.	Аналіз роботи з керівником	10.06.2022	
8.	Попередній захист роботи	15.06.2022	
9.	Коригування роботи	20.06.2022	
10.	Підготовка презентації	29.06.2022	
11.	Захист роботи	05.07.2022	

Студент _____

Керівник _____

“ ___ ” _____ 20__ р.

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

ЗАТВЕРДЖУЮ
Зав.кафедри математики,
проф., д.ф-м.н., Б. В. Олійник

_____ (підпис)
„ ____ ” _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

студенту 4-го курсу, факультету інформатики
Малому Данилу Костянтиновичу
Тема Прийняття рішень в системах керування декількома запасами

Зміст ТЧ до кваліфікаційної роботи:

Зміст
Анотація
Вступ
1 Теоретична частина
2 Практична частина
Висновки
Список літератури

Дата видачі „ ____ ” _____ 2022 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Зміст

АНОТАЦІЯ.....	5
ВСТУП.....	6
Актуальність.....	6
Мета роботи.....	6
1. ТЕОРЕТИЧНА ЧАСТИНА.....	7
2. ПРАКТИЧНА ЧАСТИНА.....	9
2.1 Постановка задачі.....	9
2.2 Процес розробки.....	9
2.3 Огляд застосунку.....	10
ВИСНОВКИ.....	14
СПИСОК ЛІТЕРАТУРИ.....	15

АНОТАЦІЯ

В цій роботі розглядаються системи керування декількома запасами, розрахунок очікуваних витрат стратегії та знаходження оптимальної стратегії в таких в таких системах. В результаті роботи було створено програмний застосунок для знаходження оптимальної стратегії в системах керування декількома запасами.

Ключові слова: система керування декількома запасами, середня вартість стратегії, оптимальна стратегія, оптимальне порогове значення.

ВСТУП

1.1 Актуальність

Тема оптимального керування запасами досліджувалась в роботах Джонсона Д.В. [1], Чорнея Р.К.[2], Кнопова П.С.[3], Беляєва Ю. А.[4], Демченко С. С.[3] та багатьох інших вчених. Гарним прикладом актуальності теорії керування для сьогодняшнього виробництва, є її застосування Джонсоном для ВПС Канади, тоді вони змогли збільшити боєготовність літаків на 24% при скороченні витрат на 300 тисяч доларів [1].

Об'єктом дослідження даної роботи є системи керування декількома запасами.

Предметом дослідження є стратегії в системах керування декількома запасами.

1.2 Мета Дослідження

Метою дослідження є створення програмного застосунку для знаходження оптимальної стратегії керування декількома запасами.

Розділ 1. Теоретична частина

Ця частина є реферативною і стосується роботи [5]

Розглянемо систему керування товарами на складі. В даному випадку значення запасів, попиту та замовлення товарів будуть дійсними і невід'ємними, що дозволить використовувати неперервні функції розподілу випадкових величин. Кожен з товарів матиме свій максимум запасу Q_i . Тут максимальна кількість товарів може залежати від величини складу, або строку зберігання самих товарів. Нехай початковий розподіл кожного з товарів є випадковою величиною. В кожен момент часу n (де $n \in N$) буде прийматися рішення про замовлення кожного з товарів. Нехай поточна кількість певного товару на складі x_i^n (де $x_i^n \in R_+$), а кількість дозамовленого товару a_i^n , де i – це номер товару, а n – поточний момент часу, тоді $a_i^n \in [0, Q_i - x_i^n]$, також в кожен момент часу ми маємо попит на кожен з товарів ξ_i^n . Врахуємо, що якщо попит перевищить кількість наявного на складі товару, тоді $x_i^n = 0$, а кількість невдоволених вимог буде врахована як дефіцит (маємо систему з втратою незадоволених вимог). Знаючи це, можемо скласти функцію обрахунку кількості товару для x_i^{n+1} :

$$x_i^{n+1} = \begin{cases} x_i^n + a_i^n - \xi_i^n, & \text{якщо } x_i^n + a_i^n \geq \xi_i^n \\ 0, & \text{якщо } x_i^n + a_i^n < \xi_i^n \end{cases}$$

Функція розподілу попиту в кожний момент часу є неперервною та не залежить від попередніх значень попиту.

Далі нам потрібно врахувати фактори, що будуть впливати на наші витрати, а саме: вартість зберігання товару, вартість замовлення товару, та збитки в наслідок дефіциту товару. Позначимо ці функції як: $C_i^1(x_i)$ – витрати на зберігання товару, $C_i^2(a_i)$ – витрати на замовлення товару, $C_i^3(y_i)$ – витрати на дефіцит товару, де y_i кількість не задоволеного попиту в певний момент часу.

Всі ці функції є монотонно неспадними та невід'ємними і задовольняють умови для таких функцій з пункту 2.3 .

На основі цих функцій можемо скласти функцію очікуваних витрат для продукту і за один період часу (якщо не було замовлення):

$$r(x_i, a_0) = C_i^1(x_i) + \int_{x_i}^{+\infty} C_i^3(y_i - x_i) dG_i(y_i)$$

Якщо ж було замовлення ($a_i > 0$):

$$r(x_i, a_i) = C_i^1(x_i) + C_i^2(a_i) + \int_{x_i+a_i}^{+\infty} C_i^3(y_i - (x_i + a_i)) dG_i(y_i)$$

Нам потрібно вибрати оптимальну стратегію для керування продуктом δ_i^* з множини усіх стратегій для даного продукту δ_i . Загальною ж, оптимальною стратегією керування усіма продуктами буде $\delta^* = \{\delta_1^*, \delta_1^*, \dots, \delta_m^*\}$, де m – це кількість продуктів.

Оскільки виконуються умови з теорем 2.3-2.5 пункту 2.3, то існує така оптимальна стратегія для керування запасами δ_i^* , $i = \overline{1, m}$ для кожного продукту:

$$\varphi(x_i \delta_i^*) = \inf_{\delta_i \in \mathfrak{R}_i} \varphi_i(x_i \delta_i)$$

де φ_i – це середня очікувана вартість стратегії δ_i , а \mathfrak{R}_i – допустимі стратегії для продукту i

$$\varphi(x_i \delta_i) = \limsup_{n \rightarrow \infty} \frac{1}{n+1} E_x^\delta \sum_{k=0}^n r(x_i^k a_i^k)$$

Основним для знаходження оптимальної стратегії є знаходження оптимального порогового значення x_i^* , такого що:

$$\delta_i^* = \begin{cases} Q_i - x_i, & \text{де } x_i < x_i^* \\ 0, & \text{де } x_i \geq x_i^* \end{cases}$$

Тобто якщо кількість товару на складі в певний момент часу стає меншою ніж x_i^* то виконується дозамовлення до максимальної кількості товару

Розділ 2. Практична частина

2.1 Постановка задачі

Опираючись на теоретичну частину, використаємо ітераційний алгоритм для пошуку оптимального порогового значення кожного товару x_i^* . В кожній ітерації ми будемо розраховувати φ_i (середню очікувану вартість для кожної стратегії), після кожної ітерації ми будемо запам'ятовувати значення φ_i для стратегії. Після цього, ми знайдемо стратегію з найменшою очікуваною вартістю, тоді число товару, при якому і відбулося дозамовлення в даній стратегії і буде найближчим до оптимального. Чим більше ітерацій зробить алгоритм, тим ближче він буде до оптимального значення x_i^* , а отже і до оптимальної стратегії. Розроблена програма має бути зручною в використанні для прорахунку витрат малого бізнесу (невеликих магазинів, кіосків, тощо).

В якості розподілу попиту будемо використовувати експоненціальний розподіл. Також важливою є можливість коригувати розподіл попиту для кожного товару окремо. Адже кожен товар має власний попит і використовувати однаковий розподіл для всіх товарів було б не коректно.

Функції витрат на зберігання, замовлення та дефіцит мають бути хоча б частково контрольовані користувачем та задаватися окремо для кожного продукту. Кількість максимального запасу кожного продукту задається окремо. Товари та обраховані для них оптимальні порогові значення мають зберігатися в пам'яті пристрою

2.2 Процес розробки

Для розробки мобільного застосунку було обрано платформу Android, щоб охопити як найбільшу кількість користувачів. У момент розробки, андроїд платформа займала близько 70% від усіх мобільних платформ на ринку.

Архітектурою для додатку було обрано MVVM, як надійний та перевірений часом підхід, рекомендований компанією Google (власне компанія яка створила Android платформу). Також були застосовані такі архітектурні підходи як Shared ViewModel та Repository. Між двома мовами нативної Android розробки (Java і Kotlin), вибір було зроблено в сторону Kotlin, через вищу швидкість написання коду цією мовою та трохи ширший функціонал. Для зберігання інформації про товари було обрано базу даних SQLite, а для більш зручного доступу до бази даних було використано бібліотеку RoomDb. Через велику кількість складних обчислень та наявність операцій по доступу до бази даних з'явилась потреба в багатопотоковій роботі програми. Тому для реалізації багатопотоковості було обрано бібліотеку Kotlin Coroutines. Також було використано бібліотеки: MaterialDesign – для дизайну інтерфейсу та Junit4 – для тестування програми під час розробки.

2.3 Огляд застосунку

Ввійшовши в застосунок бачимо основний екран, на якому відображено всі наявні товари, та розраховані на них порогові значення (якщо такі розрахунки проводились). На кожному товарі маємо кнопки для розрахунку оптимальної кількості товару та видалення товару. Справа знизу маємо кнопку додавання нового товару. Через складність обчислень, знаходження порогового значення може займати певний час після натискання кнопки. Але є можливість проводити ці обчислення паралельно для кількох продуктів, натиснувши на значок обчислень на кожному з них.

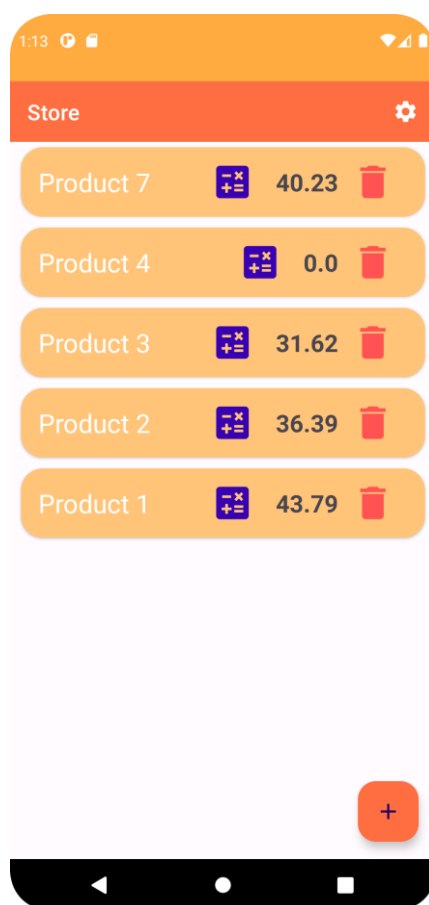


Рисунок 1. Головний екран

Натиснувши на кнопку додавання товару перейдемо на екран, де можемо додати новий товар, та налаштувати його параметри. Для кожного товару потрібно додати його ім'я, максимальну кількість товару цього виду, який може зберігатись на складі, параметер λ (інтенсивність) експоненційного розподілу попиту на товар, функцію витрат на замовлення товару та її параметри (не задані параметри будуть рахуватися як 0), функцію витрат на дифіцит та її параметри і функцію витрат на зберігання та її параметрит. Після введення всіх даних треба натиснути кнопку “Add Product” щоб додати товар.

Рисунки 2-3, Екран додавання товару

Розглянемо декілька ситуацій, щоб впевнитися, що програма працює. Для першого товару (Рисунок 4) задамо функцію витрат на дефіцит: $10x^2 + 5x$, - яка буде зростати значно швидше за функції витрат на зберігання та замовлення товару. Очікуваною поведінкою тут буде те, що програма визначить досить високе порогове значення для цього товару, щоб максимально зменшити витрати на дефіцит. Для другого товару (Рисунок 5) ж визначимо функцію дефіциту яка зростає з значно меншою швидкістю відносно функції витрат на замовлення. Очікуваною поведінкою тут буде порогове значення менше за перший випадок. І для третього товару (Рисунок 6) визначимо функцію витрат на замовлення як: $10x^2 + 5x + 5$, яка зростає значно швидше за інші функції витрат, тут очікуваною поведінкою буде порогове значення яке рівне 0, адже через високі витрати на замовлення товару, буде значно дешевше просто покривати витрати на дефіцит, не замовляючи товар.

1:13

← Add Product

Product Name
Product 1

Maximum storage amount
50

Intensity
0.2

Order Expenses
 $a*x+b$

a: 4, b: 3, c: c

Deficit Expenses
 $a*x^2+b*x+c$

a: 10, b: 5, c: c

Storage Expenses
 \sqrt{x}

a: a, b: b, c: c

Рисунок 3

1:13

← Add Product

Product Name
Product 2

Maximum storage amount
50

Intensity
0.2

Order Expenses
 $a*x+b$

a: 1, b: 2, c: c

Deficit Expenses
 $a*x+b$

a: 2, b: 5, c: c

Storage Expenses
 $a*x+b$

a: 0.2, b: 1, c: c

Рисунок 4

1:13

← Add Product

Product Name
Product 3

Maximum storage amount
50

Intensity
0.2

Order Expenses
 $a*x^2+b*x+c$

a: 10, b: 5, c: 5

Deficit Expenses
 $a*x^2+b*x+c$

a: 1, b: b, c: c

Storage Expenses
 \sqrt{x}

a: a, b: b, c: c

Рисунок 5

Розрахувавши порогові значення для кожного з товарів маємо такі значення (Рисунок 6). Як бачимо, порогове значення першого товару значно більше за порогове значення другого товару, через високу вартість дефіциту. А у третього товару порогове значення дорівнює 0, через високу вартість замовлення, буде дешевше просто покривати витрати на дефіцит. Отже у всіх трьох випадках програма прийняла вірні рішення.

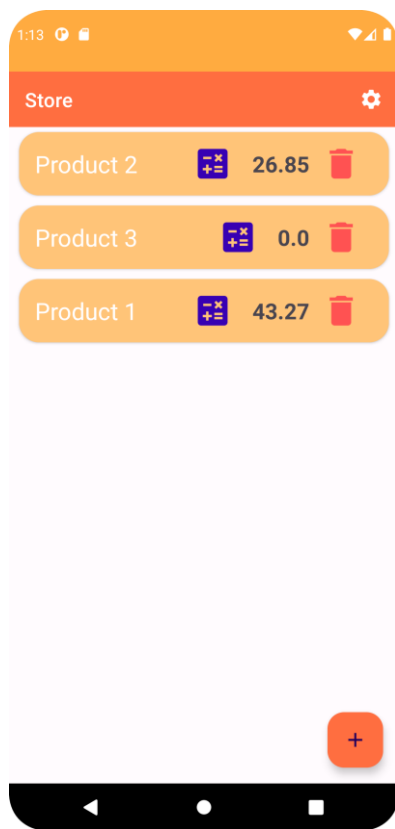


Рисунок 6

Висновки

В даній роботі було розглянуто систему керування декількома запасами та на основі відповідної теорії створено програмний застосунок для оптимального керування запасами.

СПИСОК ЛІТЕРАТУРИ

1. Johnson J. W. On stock selection at spare parts stores sections / J. W. Johnson.
// Naval Research Logistics Quarterly. – 1962 p.
2. Оптимальні стратегії для напівмарківської системи запасів (з рос.) / С. С. Демченко, П. С. Кнопов, Р. К. Чорней. // Кібернетика і системний аналіз. – 2002. – №1. – С. 146–160
3. Оптимальні стратегії для систем управління запасами с опуклою функцією витрат (з рос.) / С. С. Демченко, А. П. Кнопов, В. А. Пепеляев. // Кібернетика и системний аналіз. – 2000 p.
4. Дефіцит, ринок и управління запасами (з рос.) / Ю. А. Беляев. – 1991 p. – 228 с.
5. Оптимальні стратегії для багатопродуктових моделей керування запасами / Проценко І.Ю. // Київський національний університет імені Тараса Шевченка – 2016 p.