

Міністерство освіти і науки України  
Кам'янець-Подільський національний університет імені Івана Огієнка  
Фізико-математичний факультет  
Кафедра комп'ютерних наук

### **Кваліфікаційна робота магістра**

**з теми: «Інтелектуальна модель системи підтримки прийняття рішень для оптимізації виробництва в хлібопекарні»**

Виконав: здобувач вищої освіти групи Кп1-М24

спеціальності 122 Комп'ютерні науки

**Сідельський Михайло Леонідович**

Керівник: **Моцик Ростислав Васильович**,  
кандидат пед. наук, доцент

Рецензент: **Сморжевський Юрій Людвігович**,  
кандидат пед. наук, доцент

Кам'янець-Подільський – 2025 р.

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ХЛІБОПЕКАРСЬКОГО ВИРОБНИЦТВА ЯК ОБ'ЄКТА ОПТИМІЗАЦІЇ .....	7
1.1. Аналіз предметної області: особливості оперативного планування на хлібопекарських підприємствах .....	7
1.2. Огляд існуючих математичних моделей та інформаційних систем для оперативного планування .....	11
1.3. Постановка задачі оптимізації та архітектура Інтелектуальної системи підтримки прийняття рішень .....	15
Висновки до Розділу 1 .....	18
РОЗДІЛ 2. РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА ПРОГРАМНОЇ СИСТЕМИ.....	20
2.1. Розробка моделі прогнозування попиту на хлібобулочні вироби (Модель «Кришталева Куля») .....	20
2.2. Модель багатоцільової оптимізації виробничого графіку (Модель «Ідеальний планувальник») .....	22
2.3. Технологічна реалізація системи: архітектура, програмні засоби та база даних .....	24
РОЗДІЛ 3. ВПРОВАДЖЕННЯ, ВАЛІДАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ.....	27
3.1. Методика валідації моделей та проведення пілотного А/В тестування.....	27
3.2. Навчання персоналу, управління змінами та процедура повномасштабного впровадження .....	31
3.3 Створення веб застосунку.....	36
Висновки.....	52
Список використаних джерел.....	54

## ВСТУП

Сучасний ринок хлібопекарської продукції в Україні характеризується високою конкуренцією, динамічними змінами споживчого попиту та необхідністю жорсткого контролю собівартості. Хліб та хлібобулочні вироби є товарами з критично коротким терміном придатності, що вимагає від виробників переходу до моделі виробництва «точно в строк» (Just-in-Time). Будь-яка похибка у плануванні — чи то надмірне виробництво (що призводить до списання та фінансових втрат), чи то дефіцит продукції (що спричиняє втрату лояльності клієнтів) — безпосередньо впливає на економічну ефективність підприємства.

Традиційні методи планування, що базуються на історичних даних та евристичних правилах, виявляються недостатньо гнучкими для оперативного реагування на такі фактори, як раптові зміни погодних умов, маркетингові акції, державні свята або локальні події, які суттєво впливають на добовий попит. Це призводить до значних операційних втрат.

Актуальність обраної теми посилюється складністю виробничого процесу, зокрема, на етапі випічки, який є найбільш енергоємним та критичним з точки зору якості кінцевого продукту. На виробництві, де одночасно експлуатуються печі з різними технічними та енергетичними характеристиками, такими як MIWE ideal e+ (з використанням термального масла, висока енергоефективність, ідеальні для великих обсягів) та MIWE electro (електричні, швидше реагування, часто використовуються для спеціалізованих виробів або малих партій), задача оптимізації набуває багатофакторного та нелінійного характеру. Необхідно одночасно враховувати технологічні обмеження (час випічки, температурні режими, інерційність печей), енергетичні витрати (різна вартість енергоносіїв та пікові навантаження), потребу ринку (мінімізація часу між випічкою та відвантаженням для забезпечення максимальної свіжості).

Таким чином, центральна проблема, що підлягає вирішенню, полягає у розробці інтелектуального інструментарію, здатного автоматично генерувати

оптимальний розклад завантаження різнотипного хлібопекарського обладнання на основі динамічного прогнозу попиту та з урахуванням мінімізації загальних операційних витрат.

Наукова значущість роботи полягає у розробці та теоретичному обґрунтуванні гібридної архітектури інтелектуальної системи підтримки прийняття рішень (ІСППР), яка комбінує моделі машинного навчання (наприклад, архітектури RNN/LSTM) для точного прогнозування нелінійного та стохастичного попиту на хлібобулочні вироби.

Алгоритми комбінаторної оптимізації (зокрема, елементи генетичних або мурашиних алгоритмів) для вирішення задачі складання розкладу (Scheduling Problem) із множиною жорстких та м'яких обмежень.

Пропонований підхід дозволяє перетворити якісні знання (досвід пекарів, технологічні карти) та кількісні дані (історія продажів, споживання енергії) у формалізовану математичну модель, яка може бути ефективно вирішена за допомогою сучасних обчислювальних методів.

Практична значущість роботи підтверджується очікуваними економічними ефектами від впровадження розробленої системи на реальному виробництві. А саме зниження виробничих втрат (списань), які за рахунок підвищення точності прогнозування та більш гнучкого планування, орієнтованого на фактичний попит. Економія енергоресурсів, де оптимізація послідовності завантаження печей (особливо MIWE ideal e+, які мають високу інерційність та вимагають мінімізації простоїв), а також раціональне використання печей MIWE electro у моменти низьких тарифів або пікових потреб. Підвищення операційної ефективності, а саме скорочення часу на рутинне планування та можливість швидкої реконфігурації виробничого розкладу у режимі реального часу при виникненні непередбачуваних подій (наприклад, затримка сировини або поломка обладнання). Покращення якості продукції, для забезпечення максимальної свіжості продукції шляхом мінімізації часу між випічкою та відвантаженням.

Таким чином, розробка та впровадження ІСППР для динамічного планування завантаження різнотипного хлібопекарського обладнання є актуальною науково-технічною задачею, вирішення якої забезпечить підприємству значну конкурентну перевагу на ринку.

**Метою** дослідження є наукове обґрунтування та розробка методів, моделей і алгоритмів синтезу Інтелектуальної системи підтримки прийняття рішень (ІСППР), яка забезпечить оптимальне оперативне планування та управління хлібопекарським виробництвом за критерієм мінімізації загальних операційних витрат при максимально можливому задоволенні ринкового попиту.

**Об'єктом** дослідження є процес оперативного планування та управління хлібопекарським виробництвом, що характеризується наявністю різнотипного теплового обладнання (печі MIWE ideal e+, MIWE electro) та залежністю від динамічного, стохастичного ринкового попиту.

**Предметом** дослідження є методи, моделі та алгоритми синтезу інтелектуальної системи підтримки прийняття рішень (ІСППР) для формування оптимального розкладу завантаження гетерогенних хлібопекарських печей, що базується на точних моделях прогнозування попиту та критеріях мінімізації загальних операційних витрат.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

- проаналізувати існуючі підходи до оперативного планування в гетерогенних виробничих системах та обґрунтувати необхідність створення ІСППР.

- розробити точну прогностичну модель ринкового попиту на хлібопекарську продукцію.

- розробити математичну модель та алгоритми оптимізації для формування оптимального розкладу завантаження гетерогенних теплових печей (MIWE ideal e+, MIWE electro).

- визначити функціональну та структурну архітектуру інтелектуальної системи підтримки прийняття рішень, включаючи модулі збору даних, прогнозування, оптимізації та візуалізації результатів планування.
- розробити програмний прототип ІСППР, реалізувати в ньому запропоновані моделі й алгоритми
- провести експериментальні дослідження для підтвердження їхньої ефективності та порівняння з традиційними методами планування.

Наукова новизна отриманих результатів полягає в удосконаленні математичної моделі оперативного планування хлібопекарського виробництва шляхом інтеграції гетерогенних виробничих потужностей (теплових печей із різними технічними параметрами та обмеженнями) у єдиний критерій оптимізації, що враховує мінімізацію загальних операційних витрат (енергія, час) за умов динамічного ринкового попиту. Розробці структурно-функціональної архітектури Інтелектуальної системи підтримки прийняття рішень (ІСППР) для хлібопекарського виробництва, що відрізняється модульністю, інтеграцією прогностичного та оптимізаційного блоків та адаптивним інтерфейсом, орієнтованим на оперативне коригування виробничих рішень в умовах реального часу.

Практична значущість одержаних результатів полягає у тому, що впровадження розробленої ІСППР дозволить зменшити операційні витрати підприємств хлібопекарської промисловості на 10-15% (за рахунок оптимізації завантаження обладнання та мінімізації енергоспоживання), а також знизити втрати від нереалізованого попиту (на 5-8%) завдяки більш точному прогнозуванню. Розроблений програмний прототип ІСППР є готовим інструментом для менеджерів, який забезпечує швидке формування оптимальних виробничих планів (розклад завантаження печей) у відповідь на оперативні зміни попиту або виробничих умов.

Структура кваліфікаційної роботи.

# **РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ХЛІБОПЕКАРСЬКОГО ВИРОБНИЦТВА ЯК ОБ'ЄКТА ОПТИМІЗАЦІЇ**

## **1.1. Аналіз предметної області: особливості оперативного планування на хлібопекарських підприємствах**

Оперативне планування виробництва є критично важливим етапом управління промисловим підприємством, особливо в умовах масового, безперервного та багатомономенклатурного виробництва, яким є хлібопекарська промисловість. Процес виготовлення хлібобулочної продукції характеризується чіткою послідовністю технологічних операцій та високою чутливістю до часових та температурних параметрів.

Розглянемо ключові технологічні етапи виробництва хліба. Основний етап приготування сировини та заміс тіста. Цей етап включає дозування інгредієнтів та замішування тіста. Час, витрачений на заміс та подальшу ферментацію, безпосередньо впливає на якість кінцевого продукту.

Наступний це розділка тіста. Процес поділу великої маси тіста на порційні заготовки, що супроводжується вторинною ферментацією. За ним йде випікання, найбільш енергоємний і критичний етап, що вимагає використання печей з різними технічними характеристиками (продуктивність, енергоспоживання, температурні режими). Наступним етапом йде охолодження та пакування. Фінальні стадії, необхідні для підготовки продукції до транспортування.

На відміну від стратегічного (довгострокового) та тактичного (середньострокового) планування, оперативне планування (або диспетчеризація) фокусується на узгодженні виробничих завдань у короткостроковому періоді (від зміни до доби). Його головна мета – скласти оптимальний графік роботи виробничих ділянок та обладнання, який забезпечить виконання прогнозованого замовлення за мінімальних витрат часу та ресурсів, а також дотримання критичних технологічних термінів.

Ефективність оперативного планування в хлібопеченні визначається необхідністю одночасного обліку значної кількості жорстких та динамічних обмежень. Часові та технологічні обмеження – час ферментації. Процес бродіння тіста не може бути прискорений або уповільнений довільно, оскільки це порушує якісні характеристики продукту. Це створює жорсткі часові "вікна" для подальших операцій.

Час переналагодження (Set-up Time), зміна номенклатури продукції на одній лінії або печі вимагає часу на переналагодження, що знижує загальну ефективність. Урахування мінімального часу простою між партіями є критичним для оптимізації.

Проблема гетерогенності виробничих потужностей (печей) є одним із найсуттєвіших викликів для хлібопекарських підприємств. Технологічний парк, як правило, включає печі різних поколінь та типів. Кожна піч має унікальну максимально допустиму кількість виробів за одиницю часу.

Споживання електроенергії, газу чи пари може значно відрізнятись, що прямо впливає на операційні витрати, особливо при динамічному тарифі. Деякі печі можуть бути налаштовані лише для випікання певних видів продукції (наприклад, житній хліб проти багетів). Система має не просто розподілити замовлення, а призначити кожену партію на найбільш економічно вигідну піч, враховуючи її доступність та енергоефективність для конкретної номенклатури.

Наявність борошна, дріжджів та спеціальних інгредієнтів у виробничих бункерах на початок зміни є лімітуючим фактором. Оперативний план повинен бути реалізований лише в межах фізичної наявності сировинних ресурсів.

На відміну від багатьох інших галузей, хлібопекарське виробництво працює в умовах надзвичайно високої волатильності попиту та ультракороткого терміну придатності продукції. Ці фактори перетворюють планування на задачу високого ризику. Чинники, що впливають на попит.

Попит на хлібобулочні вироби є високочутливим до календарних чинників, таких як вихідні дні, святкові періоди (Різдво, Великдень), шкільні канікули.

Різке зниження або підвищення температури, опади можуть впливати на купівельну спроможність та логістичні ланцюжки. Маркетингові акції конкурентів, знижки в торгових мережах. Традиційні методи прогнозування попиту часто виявляються недостатньо точними, що призводить до системних помилок у плануванні. Це обґрунтовує необхідність застосування інтелектуальних методів машинного навчання, здатних врахувати складні, нелінійні взаємозв'язки між цими факторами.

Короткий термін придатності хліба (як правило, 24–72 години) ставить планувальника перед дилемою. Виготовлення продукції понад реальний попит призводить до повернень із торгових мереж (забракована продукція) та прямих фінансових втрат. Недостатня кількість продукції на полицях призводить до втрати потенційного прибутку та зниження лояльності клієнтів (Stock-out cost). Мета оперативного планування полягає у мінімізації сукупних втрат від цих двох протилежних ризиків. Точність прогнозування, розробленого в ІСППР, стає ключовим інструментом для досягнення цього балансу.

Аналіз предметної області дає змогу чітко сформулювати, що саме підлягає оптимізації, та які показники будуть використовуватись для оцінки ефективності. Об'єктом управління є процес розподілу замовлень на виробництво (партій продукції) між гетерогенними виробничими потужностями (печами) в межах часового горизонту (змiна/доба).

Визначення вхідних змінних (Input Data):

$P_j$  – Прогнозований попит (кількість одиниць продукції  $j$ );

$T_i$  – Часовий профіль доступності печі  $i$  (години роботи);

$E_{ij}$  – Енергоспоживання печі  $i$  для випікання одиниці продукції  $j$  (залежність, що враховує гетерогенність);

$C_{raw}$  – Наявність критичної сировини.

Основне завдання оперативного планування хлібопекарського підприємства зводиться до задачі мінімізації загальних операційних витрат  $Z$ :

$$\min Z = \min \sum_i + \sum_j (C_{\text{prod}} X_{ij}) + \sum_j (C_{\text{over}} O_j) + \sum_j (C_{\text{under}} U_j)$$

де:

$X_{ij}$  – Кількість продукції  $j$ , виробленої на печі  $i$ . Це єдина цілочисельна змінна рішення.

$C_{\text{prod}}$  – Вартість виробництва одиниці продукції (включаючи енерговитрати).

$O_j$  – Обсяг надвиробництва (Overproduction) продукції  $j$ .

$U_j$  – Обсяг дефіциту (Underproduction) продукції  $j$ .

$C_{\text{over}}$  та  $C_{\text{under}}$  – Штрафні коефіцієнти за надвиробництво та дефіцит відповідно.

Розроблена ІСППР повинна задовольняти наступні ключові обмеження. Обмеження потужності печі – загальний обсяг виробництва на печі  $i$  не може перевищувати її максимальної продуктивності за зміну.

Обмеження попиту (баланс виробництва). Сумарний обсяг виготовленої продукції повинен збалансувати прогнозований попит, включаючи відхилення:

$$\sum_i X_{ij} + U_j - O_j = P_j$$

Дотримання технологічних часових вікон та мінімальних часів переналагодження.

Таким чином, розробка Інтелектуальної системи підтримки прийняття рішень є обґрунтованою з погляду необхідності вирішення складної багатофакторної задачі оптимізації завантаження гетерогенних потужностей в умовах високодинамічного та ризикованого попиту, що не може бути ефективно вирішене традиційними методами.

## 1.2. Огляд існуючих математичних моделей та інформаційних систем для оперативного планування

Оперативне планування виробництва є однією з найстаріших і найбільш досліджених проблем в галузі операційного менеджменту. Історично для вирішення цієї задачі застосовувалися такі класичні моделі, як модель економічно обґрунтованого розміру замовлення (EOQ – Economic Order Quantity). Вона використовується для визначення оптимального розміру виробничої партії з метою мінімізації сукупних витрат на зберігання запасів та переналагодження обладнання. Хоча модель корисна для стратегічного управління запасами сировини, вона малоефективна для оперативного планування готової продукції з ультракоротким терміном придатності.

Теорія масового обслуговування (Queuing Theory), яка дозволяє аналізувати потоки роботи та оптимізувати завантаження робочих центрів, зменшуючи час очікування. Вона може бути застосована для балансування швидкості замісу тіста та пропускної спроможності печей.

Недостатність стандартних ERP/MRP систем. Системи планування потреб у матеріалах (MRP, Manufacturing Resource Planning) та їхні інтегровані послідовники (ERP, Enterprise Resource Planning) є основою управління на більшості підприємств. Вони чудово справляються з функціями управління запасами та основними даними (BOM – Bill of Materials). Формування календарного плану виробництва (Capacity Planning). Фінансового та бухгалтерського обліку.

Проте, стандартні модулі MRP II/ERP мають суттєві обмеження щодо оперативного планування хлібопекарського виробництва. Принцип нескінченної потужності, де більшість MRP-систем спочатку припускають необмежену доступність ресурсів і лише потім перевіряють їхню наявність, що часто призводить до нереалістичних графіків. Відсутність тонкої оптимізації, де стандартні системи використовують прості правила пріоритетів (наприклад, FCFS — First-Come, First-Served) замість складних

алгоритмів, які можуть мінімізувати енерговитрати або час переналагодження. Вони не здатні врахувати гетерогенність печей та динамічні тарифи.

Для вирішення задач оперативного планування, які виходять за рамки можливостей ERP-систем, застосовуються складні математичні моделі, що дозволяють формалізувати цільову функцію та обмеження.

1. Лінійне програмування (ЛП) та цілочисельне лінійне програмування (ЦЛП). Ця модель ЦЛП є найбільш поширеним підходом для оптимізації розподілу ресурсів. Вона дозволяє точно вирішити задачу мінімізації витрат або максимізації прибутку, як це було формалізовано в параграфі 1.1.

Вона гарантує знаходження глобально оптимального рішення (за умови його існування). З основних недоліків – зі зростанням кількості змінних (кількість партій, печей, номенклатури та часових періодів) задача стає NP-складною. Обчислювальний час для великих виробництв може перевищувати час, доступний для оперативного прийняття рішень (кілька хвилин).

2. Класичні задачі календарного планування (Scheduling). Процес складання графіка роботи обладнання зазвичай моделюється як одна з відомих задач. Job-Shop Scheduling (JSS) – найбільш близька модель до хлібопечення, де кожен вид продукції проходить свій унікальний маршрут (заміс, ферментація, випікання), а обладнання (печі, міксери) є гетерогенним і може обробляти різні замовлення. Flow-Shop Scheduling (FSS) – менш релевантна, оскільки передбачає однаковий технологічний маршрут для всіх виробів, що не відповідає різноманіттю номенклатури хлібопекарні.

Основною проблемою JSS є необхідність одночасного обліку залежностей між операціями (тісто має бути готове до випікання) та обмежень потужності обладнання. У реальному виробництві, де потрібно врахувати час переналагодження, динамічні зміни замовлень і вартість енергії, ці задачі швидко стають нерозв'язними для точних алгоритмів за прийнятний час.

Виявлені обчислювальні проблеми ЦЛП та JSS в реальному часі вимагають використання евристичних (наближених) та метаввристичних

алгоритмів. Ці методи не гарантують абсолютного оптимуму, але дозволяють знайти дуже близьке до нього рішення за прийнятний час.

Розглянемо генетичні алгоритми (ГА, Genetic Algorithms), це найпоширеніший інструмент для вирішення складних задач календарного планування. ГА базуються на принципах біологічної еволюції та природного добору. Його механізм роботи, генерується початкова множина можливих графіків (хромосом). Кожен графік оцінюється за цільовою функцією (наприклад, мінімізація витрат та штрафів). Найкращі графіки комбінуються (схрещування) та модифікуються (мутація), щоб сформувати наступне



покоління. ГА є особливо цінними, оскільки вони можуть легко інтегрувати складні, нелінійні обмеження (як-от часові вікна ферментації або унікальні енергетичні профілі печей) у фітнес-функцію. Вони здатні знаходити ефективні рішення в просторі пошуку, занадто великому для точних методів.

2. Імітаційне моделювання (Simulation). Моделювання дискретних подій (Discrete Event Simulation, DES) дозволяє оцінити ефективність розробленого плану без його фактичної реалізації. DES моделює весь процес виробництва (включаючи випадкові фактори, такі як поломки обладнання або затримки сировини) і надає детальні показники продуктивності, часу очікування та

завантаження обладнання. Він дозволяє проводити аналіз "що, якщо" (what-if analysis) для оцінки стійкості плану.

3. Нейронні мережі для прогнозування. Для вирішення проблеми волатильності попиту, виділеної в 1.1, інтелектуальні системи часто використовують рекурентні нейронні мережі (RNN) або моделі ARIMA, що дозволяють будувати точніші прогнози, інтегруючи часові ряди, метеорологічні та календарні дані. Це забезпечує більш надійну вхідну інформацію для евристичних алгоритмів планування.

На ринку представлено два основні класи інформаційних систем, які теоретично можуть вирішувати завдання планування: ERP-системи та спеціалізовані системи APS/MES.

Системи класу MES (Manufacturing Execution Systems). Системи MES призначені для оперативного управління цехом. Вони виконують функції збору даних про виробництво в режимі реального часу, контроль якості та відстеження партій (Traceability), управління роботою обладнання.

MES є чудовим засобом *виконання і моніторингу* плану, але зазвичай не містять потужних вбудованих *оптимізаційних* алгоритмів. Вони виконують графік, складений або вручну, або зовнішньою системою.

Системи класу APS (Advanced Planning and Scheduling). Системи APS (наприклад, Siemens Preactor, SAP APO) розроблені спеціально для вирішення складних задач календарного планування. Хоча вони використовують деякі згадані оптимізаційні методи (евристики), їхнє впровадження є надзвичайно дорогим. Крім того, вони часто вимагають значного доопрацювання та кастомізації для обліку унікальних, гетерогенних факторів (як-от специфічні енергопрофілі печей і жорсткі часові вікна ферментації) на конкретному підприємстві. Спроби "вписати" унікальну хлібопекарську задачу в загальний APS-шаблон часто призводять до зниження ефективності.

Висновок та обґрунтування розробки ІСППР. Проведений огляд демонструє, що існуючі рішення залишають суттєвий функціональний розрив, а саме ERP/MRP не мають необхідної математичної бази для оптимізації,

ЦЛП/JSS – обчислювально нерозв'язні за необхідний оперативний час, комерційні APS/MES – дорогі, складні у впровадженні та потребують глибокої адаптації для обліку критичної гетерогенності обладнання та мінімізації штрафів за дефіцит/надвиробництво в умовах високодинамічного попиту.

Таким чином, обґрунтовується необхідність розробки спеціалізованої Інтелектуальної системи підтримки прийняття рішень (ІСППР), яка буде інтегрувати найбільш ефективні компоненти: нейромережеве прогнозування попиту та метаевристичні алгоритми (наприклад, генетичні) для оптимізації розподілу замовлень між гетерогенними печами, мінімізуючи сукупні операційні витрати.

### **1.3. Постановка задачі оптимізації та архітектура Інтелектуальної системи підтримки прийняття рішень**

Сторінка 1. Уточнена математична постановка задачі

На основі аналізу потреб хлібопекарського підприємства (1.1) та огляду існуючих моделей (1.2), уточнюємо математичну постановку задачі оперативного планування як задачі багатоцільової мінімізації витрат з жорсткими часовими обмеженнями на гетерогенних ресурсах.

Цільова функція (мінімізація загальних витрат). Мінімізувати  $F$ :

$$F = W_{Op} + W_{Inv} + W_{Set} + W_{Pen}$$

Де  $W_{Op}$  – загальні операційні витрати (переважно електроенергія, враховуючи динамічні тарифи).

$W_{Inv}$  – витрати на зберігання запасів (залишки готової продукції).

$W_{Set}$  – витрати на переналагодження обладнання (перехід з одного виду продукції на інший).

$W_{Pen}$  – штрафи за невиконання замовлень (дефіцит) або за надлишкове виробництво.

Загальна кількість виробленої продукції  $P_i$  за період  $t$  повинна покривати прогнозований попит  $D_i$  з урахуванням початкових запасів

Обмеження потужності обладнання (Печі  $k$ ):

Сумарний час випікання  $T_{i,k,t}$  у кожній печі  $k$  за період  $t$  не повинен перевищувати її доступну потужність  $T_{k,t}$

Важливим елементом є гетерогенність  $T_{k,t}$  потужність залежить від типу продукції  $i$  та енергетичного профілю печі  $k$ .

Обмеження часових вікон ферментації (Жорсткі залежності) – час початку випікання  $t_i$  має бути жорстко прив'язаний до часу завершення замісу.

Введення бінарної змінної  $X_{i,j,k}$  (дорівнює 1, якщо піч  $k$  переналагоджується з продукту  $i$  на  $j$ ) дозволяє інтегрувати витрати  $W_{Set}$  у цільову функцію, мінімізуючи кількість та складність переналагоджень.

Для вирішення поставленої задачі, яка є NP-складною і вимагає оперативного рішення, пропонується модульна архітектура Інтелектуальної системи підтримки прийняття рішень (ІСППР). Система складається з трьох основних функціональних рівнів.

Рівень даних (Data Layer) – відповідає за збір, очищення та зберігання вхідних даних (історичний попит, тарифи на електроенергію, технологічні карти, поточний стан обладнання та запасів).

Рівень інтелектуальних модулів (Intelligent Modules Layer) – ядро системи, що містить моделі прогнозування та оптимізації.

Рівень користувацького інтерфейсу (Presentation Layer), забезпечує взаємодію з диспетчером, відображаючи кінцевий графік, ключові показники ефективності (KPI) та інструменти для ручного корегування.

Ключові компоненти ІСППР це:

Модуль прогнозування попиту (MPP), який використовує моделі машинного навчання (наприклад, LTSM, RNN) для аналізу історичних продажів, календарних факторів (свята) та зовнішніх факторів (температура, акції). Його вихід – точний прогноз попиту  $D_{i,t}$ .

Модуль оптимізації календарного плану (МОКР), який приймає дані від MPP та технологічні/ресурсні обмеження. Застосовує метаевристичні алгоритми (наприклад, генетичний алгоритм) для знаходження графіка, що мінімізує цільову функцію  $F$ .

База знань обмежень (БЗО), що зберігає жорсткі технологічні параметри (часи ферментації  $T_i$ , час випікання  $T_{\text{Bake}}$  для кожної печі  $k$ , витрати на переналагодження).

Модуль прогнозування попиту (МРР). Метою МРР є мінімізація невизначеності на вході задачі планування. Для хлібопекарні критично важливим є прогнозування попиту на короткострокову перспективу (до 24-48 годин).

Таб1. Застосування моделей часових рядів із зовнішніми регресорами (Exogenous Variables).

Вхідні дані (Змінні)	Тип даних	Вплив на попит
Історичні продажі	Часовий ряд	Базова сезонність та тренд
День тижня / Свята	Категоріальні	Піковий попит на вихідних та свята
Погодні умови (Температура)	Зовнішній регресор	Температура повітря може впливати на попит на певні види продукції
Ціна та акції	Керуючий фактор	Значний вплив на обсяг продажів

Результатом роботи МРР є вектор прогнозованого попиту  $D$  та рівень його дисперсії, що використовується МОКР для визначення оптимального страхового запасу.

Модуль оптимізації календарного плану (МОКР) є обчислювальним ядром системи, яке виконує функцію планувальника, що враховує гетерогенність обладнання. Використання генетичного алгоритму (ГА). Де кожна хромосома являє собою повний графік виробництва, що включає: послідовність виробництва для кожної печі  $k$ , час початку та завершення кожної виробничої партії. Пряме обчислення цільової функції  $F$  (сумарні витрати). ГА шукає хромосому з мінімальним значенням  $F$ .

Особливість адаптації ГА – спеціальні оператори схрещування та мутації розробляються для збереження жорстких обмежень (наприклад,

оператор мутації не може порушити часове вікно ферментації, інакше хромосома отримує високий штраф).

Наукова новизна роботи – комплексна інтеграція: Попит → Енергія → Ресурси. Вперше здійснюється інтеграція трьох ключових факторів (нечіткий попит, динамічні енергетичні тарифи та гетерогенність обладнання) в єдину цільову функцію оптимізації календарного планування. Генетичний алгоритм із спеціалізованими обмеженнями, в якому розроблено оригінальну схему кодування хромосом та адаптовані генетичні оператори, здатні ефективно обробляти жорсткі часові обмеження (наприклад, вікна ферментації) та складні нелінійні залежності. Удосконалена модель штрафних санкцій, що враховує надкороткий термін придатності хлібобулочної продукції, тим самим мінімізуючи втрати від списання. Оптимізація графіків роботи печей (найбільш енергоємного обладнання) дозволяє зміщувати пікове споживання на часові інтервали з мінімальними тарифами, забезпечуючи економію електроенергії до 15-20%. Зниження дефіциту та надлишкового виробництва за рахунок точного прогнозування та оптимізації виробничої програми.

ІСППР надає оптимальний графік протягом декількох хвилин, дозволяючи диспетчеру швидко реагувати на зміни попиту або непередбачені події (поломка обладнання, затримка сировини). Ефективне балансування завантаження гетерогенних печей, що сприяє раціональному використанню потужностей.

### **Висновки до 1 Розділу**

У першому розділі було обґрунтовано актуальність і наукову новизну розробки Інтелектуальної системи підтримки прийняття рішень (ІСППР) для оперативного календарного планування у хлібопекарській промисловості.

Встановлено, що ключовими викликами сучасного виробництва є висока енергоємність технологічного процесу, динамічні тарифи на електроенергію, гетерогенність обладнання (різні типи печей) та жорсткі часові обмеження (ферментація тіста). Необхідність мінімізації втрат від

короткого терміну придатності вимагає інтеграції точного прогнозування попиту та оптимізації виробничого графіка.

Огляд існуючих підходів до планування виявив, що більшість моделей не забезпечують комплексної інтеграції цільових функцій, які одночасно включають динамічні енергетичні витрати, витрати на переналагодження та жорсткі технологічні залежності.

Сформульовано математичну постановку задачі як багатоцільову мінімізацію загальних витрат ( $W_{Op} + W_{Inv} + W_{Set} + W_{Pen}$ ) з урахуванням жорстких обмежень потужності та часових вікон ферментації.

Розроблено модульну архітектуру ІСППР, яка включає модуль прогнозування попиту (на основі методів Machine Learning) та модуль оптимізації календарного плану. Для вирішення NP-складної задачі планування обґрунтовано застосування генетичного алгоритму, адаптованого для роботи з жорсткими часовими обмеженнями.

Запропонована архітектура та уточнена математична модель закладають основи для розробки ефективного інструменту, здатного забезпечити зниження операційних витрат підприємства на 15–20% та підвищення гнучкості виробництва.

## РОЗДІЛ 2. РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА ПРОГРАМНОЇ СИСТЕМИ

### 2.1. Розробка моделі прогнозування попиту на хлібобулочні вироби (Модель «Кришталева Куля»)

Прогнозування попиту на хлібобулочні вироби є класичною задачею аналізу часових рядів, ускладненою значною нелінійністю та високим впливом зовнішніх факторів. Висока варіативність попиту, спричинена ефектом «свіжості» продукції, сезонністю, циклічністю (день тижня) та акційною активністю, вимагає застосування гібридних, адаптивних моделей.

Аналіз вихідних даних показує, що часовий ряд попиту на хлібобулочні вироби містить чітко виражені компоненти: тренд (загальна динаміка), мультиплікативну сезонність (тижнева та річна циклічність) та залишкову випадкову складову.

Традиційні моделі (такі як ARIMA/SARIMA) є високоефективними для стаціонарних або стаціонаризованих часових рядів, але вимагають попередньої декомпозиції та чутливі до наявності значної кількості зовнішніх регресорів. З огляду на необхідність інтеграції великого масиву даних про свята та маркетингові події, перевага надається моделям, які за своєю природою є більш гнучкими до включення цих факторів.

У рамках «моделі кришталева куля» обґрунтовано вибір на користь регресійної моделі з адитивним розкладом часового ряду, наприклад, імплементації моделі Фейсбук Prophet. Основними перевагами цього підходу є гнучкість у моделюванні тренду, а саме: здатність автоматично обробляти нелінійні зміни тренду, спричинені структурними зсувами на ринку чи запуском нових асортиментних груп. Моделювання тижневої, місячної та річної сезонності без складної ручної гармонійної декомпозиції. Можливість інтегрувати зовнішні фактори як екзогенні змінні (Extra Regressors), зберігаючи при цьому основний адитивний розклад.

Цей підхід дозволяє розкласти прогнозоване значення  $Y(t)$  на його основні компоненти, як показано у формулі:

$$Y(t) = g(t) + s(t) + h(t) + \epsilon t$$

де  $g(t)$  – функція тренду,  $s(t)$  – періодична функція сезонності,  $h(t)$  – функція впливу свят та акцій (регресорів),  $\epsilon t$  – залишкова похибка.

Для підвищення точності моделі обов'язковим є включення низки екзогенних змінних, що прямо впливають на споживчу поведінку:

Календарні ефекти, використання бінарних змінних (Dummy Variables) для ідентифікації днів тижня, державних свят, передсвяткових та післясвяткових періодів. Це дозволяє моделі фіксувати різке зростання попиту напередодні вихідних чи національних свят.

Маркетингові акції, бінарні змінні, що відображають активність конкурентів або власні промоційні заходи (знижки, 2+1, спеціальні пропозиції). Дані про акції включаються до прогнозу як відома майбутня інформація.

Погодні умови, включення агрегованих метрик погоди (середня температура, ймовірність опадів) як додаткових числових регресорів, оскільки попит на певні категорії продукції (наприклад, супутні напої) є чутливим до погодних змін.

Для об'єктивної оцінки якості роботи моделі використовується валідація на відкладеному тестовому наборі даних (Out-of-Sample Testing). Ключові метрики для оцінки похибки:

Середня абсолютна похибка (MAE – Mean Absolute Error):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i|$$

Ця метрика є особливо важливою для операційного планування, оскільки вона вимірює середній абсолютний обсяг помилки в одиницях продукції (наприклад, кг або шт.). MAE легко інтерпретується виробничим персоналом як середня кількість одиниць, на яку прогноз відхилився від факту.

Середньоквадратична похибка (RMSE – Root Mean Square Error):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2}$$

RMSE є більш жорсткою метрикою, оскільки вона квадратично штрафувє великі відхилення. Зменшення RMSE є критичним для мінімізації ризику значних надлишків або дефіцитів, що напряду впливають на фінансові втрати.

Цільовою є мінімізація обох метрик. Успішною вважається модель, яка демонструє MAE не вище  $X\%$  від середньодобового попиту, забезпечуючи високу надійність вхідних даних для подальшого етапу оптимізації.

## 2.2. Модель багатоцільової оптимізації виробничого графіку

### (Модель «Ідеальний планувальник»)

Модель «Ідеальний планувальник» розроблена для перетворення прогнозованого попиту (отриманого з Моделі «Кришталева куля») на поопераційний виробничий графік, який є одночасно технологічно можливим та економічно ефективним. Завдання виробничого планування на хлібопекарському підприємстві є класичною задачею багатоцільової оптимізації (Multi-Objective Optimization), оскільки передбачає компроміс між конкуруючими цілями.

Основні цілі оптимізації можуть бути зведені до єдиної цільової функції  $Z$ , яка представляє мінімізацію сумарних витрат при задоволенні попиту. Домінуючими складовими  $Z$  є:

Мінімізація енергетичних витрат ( $C_{\text{energy}}$ ). З огляду на диференційовані тарифи (нічний тариф є значно нижчим), пріоритетом є планування найбільш енергоємних процесів (випікання) у години низьких тарифів.

Мінімізація витрат на переналагодження ( $C_{\text{setup}}$ ). Зміна типу продукції на одній лінії вимагає часу та ресурсів (чистка, калібрування). Система мінімізує кількість та складність цих переналагоджень, групуючи подібні вироби в послідовні партії.

Мінімізація витрат на зберігання та втрати ( $C_{\text{inventory}}$ ). Мінімізація як надлишкового запасу (Overstock, що веде до списання через малий Shelf Life), так і дефіциту (Out-of-Stock, що веде до втрачених продажів).

Цільова функція  $Z$  набуває вигляду:

$$\min Z = \omega_1 \sum C_{\text{energy}} + \omega_2 \sum C_{\text{setup}} + \omega_3 \sum C_{\text{inventory}}$$

де  $\omega_1, \omega_2, \omega_3$  – вагові коефіцієнти, що визначаються менеджментом підприємства для відображення стратегічного пріоритету (наприклад,  $\omega_1$  буде високим у регіонах з дорогим нічним тарифом).

Система обмежень моделі (Constraints). Для забезпечення реалістичності та технологічної можливості графіка, модель має задовольняти низку жорстких обмежень (Hard Constraints). Обмеження задоволення попиту ( $C_{\text{demand}}$ ). Кількість виробленої продукції  $P_{i,t}$  для кожного типу продукції  $i$  у часі  $t$  повинна задовольняти прогнозований попит  $\hat{Y}_{i,t}$  з урахуванням наявних запасів  $I_{i,t-1}$  та необхідного страхового запасу  $S_i$ :

$$P_{i,t} + I_{i,t-1} \geq Y_{i,t} + S_i, \quad \forall i, t$$

Обмеження виробничої потужності ( $C_{\text{capacity}}$ ), це сумарний час, необхідний для випікання всіх запланованих партій на виробничій лінії  $j$  у період  $t$ , не може перевищувати доступний робочий час  $T_{j,t}$ :

$$\sum Time_{i,j} \cdot X_{i,j,t} \leq T_{j,t} T_{\{j,t\}}, \quad \forall i, t$$

де  $X_{i,j,t}$  – кількість партій продукту  $i$  на лінії  $j$  у період  $t$ , а  $Time_{i,j}$  – час виробництва партії.

Технологічні обмеження ( $C_{\text{tech}}$ ). Встановлення мінімального та максимального часу випікання, а також правил послідовності (наприклад, на лінії з виробництва житнього хліба не можна одразу перейти до виробництва білого без повної переналагодження).

З огляду на дискретність ключових змінних (кількість партій, вибір лінії, бінарні змінні для переналагодження) та лінійність більшості обмежень та цільової функції, оптимальним методом є змішано-цілочислове лінійне програмування (MILP – Mixed-Integer Linear Programming).

Задача MILP дозволяє використовувати як неперервні (наприклад, рівень запасу), так і цілочислові/бінарні (наприклад, чи відбулося переналагодження) змінні.

Для вирішення MILP-задачі будуть задіяні спеціалізовані солвери (наприклад, інтеграція з PuLP/Pyomo з використанням GLPK, CPLEX або Gurobi). Для великих, складних задач, де точне вирішення може бути надто ресурсомістким, розглядається можливість застосування евристичних методів, таких як генетичні алгоритми або імітація відпалу, для пошуку наближених до оптимальних рішень за прийнятний час.

Таким чином, модель «Ідеальний планувальник» функціонує як складний математичний інструмент, що балансує між різними економічними факторами та фізичними обмеженнями для генерації найбільш ефективного виробничого графіка.

### **2.3. Технологічна реалізація системи: архітектура, програмні засоби та база даних**

Ефективна робота ІСППР вимагає надійної, масштабованої та модульної архітектури. Система реалізується за принципом багаторівневої архітектури (Multi-Tier Architecture), що забезпечує чітке розділення відповідальності між компонентами (моделі даних, бізнес-логіка та користувацький інтерфейс).

Архітектура системи. Система складається з трьох основних логічних рівнів.

Рівень даних (Data Layer) відповідає за збір, зберігання та первинну обробку всієї необхідної інформації. Включає реляційні та нереляційні бази даних.

Рівень бізнес-логіки та моделювання (Application/Modeling Layer) – це ядро системи, де виконуються модель «Кришталева куля» (прогнозування) та модель «Ідеальний планувальник» (оптимізація). Тут також розташовані алгоритми підготовки даних (ETL) та інтерпретації результатів.

Рівень користувацького інтерфейсу (Presentation Layer) забезпечує взаємодію користувача із системою, включаючи введення параметрів, візуалізацію прогнозу, відображення оптимізованого графіку та генерацію звітів.

Обмін даними між рівнями відбувається за допомогою стандартизованих інтерфейсів (API), що дозволяє незалежно оновлювати та масштабувати кожен рівень.

Для реалізації двох ключових моделей та підтримки ІСППР обрано наступний стек технологій.

Компонент системи	Засіб / технологія	Обґрунтування вибору
Розробка Моделей (ML/Optimization)	Python 3.x	Стандартизований інструмент для наукових обчислень. Доступність високопродуктивних бібліотек.
Прогнозування (Модель 2.1)	Prophet (Meta), Scikit-learn, Statsmodels	Prophet — для гнучкого моделювання сезонності та тренду. Scikit-learn — для роботи з екзогенними регресорами.
Оптимізація (Модель 2.2)	PuLP / Pyomo	Бібліотеки для моделювання та вирішення задач лінійного та змішано-цілочислового програмування (MILP). Інтеграція з відкритими солверами (наприклад, GLPK).
База Даних (Рівень 1)	PostgreSQL	Реляційна СУБД, що підтримує складні транзакції, забезпечення цілісності даних (ACID) та високу швидкість обробки часових рядів.
Користувацький Інтерфейс (Рівень 3)	React / Streamlit	React для створення повнофункціонального, інтерактивного веб-інтерфейсу.

		Streamlit для швидкого прототипування та візуалізації результатів моделей.
--	--	--

Використання контейнеризації (Docker) є критичним для забезпечення відтворюваності, ізоляції та швидкого розгортання ІСППР на різних виробничих майданчиках.

База даних має модульну структуру для ефективного зберігання та агрегації різнорідних даних, необхідних для моделювання та роботи системи.

**1. Таблиця історичного попиту (Fact\_Demand):**

date\_key (Дата, первинний ключ)

product\_id (Ідентифікатор продукції)

quantity\_sold (Фактично продана кількість)

selling\_price (Ціна продажу)

**2. Таблиця екзогенних факторів (Dim\_Exogenous):**

date\_key

is\_weekend (Бінарна: 0/1)

is\_holiday (Бінарна: 0/1, Календар свят)

avg\_temperature (Середня температура в регіоні)

promo\_type (Тип маркетингової акції: None/Discount/2+1)

**3. Таблиця виробничих обмежень (Dim\_Capacity):**

line\_id (Ідентифікатор лінії)

product\_id (Зв'язок з продукцією)

time\_per\_batch (Час виробництва партії)

setup\_time (Час переналагодження)

max\_daily\_capacity (Максимальна добова потужність)

**4. Таблиця результатів оптимізації (Output\_Schedule):**

date\_key

product\_id

optimized\_batch\_count (Кількість партій, рекомендована системою)

start\_time\_plan (Плановий час початку виробництва)

cost\_z (Значення цільової функції для цього рішення)

Така структура забезпечує ефективне агрегування історичних продажів з відповідними зовнішніми та виробничими умовами, що є необхідною передумовою для навчання моделі «Кришталева куля» та подальшої оптимізації.

## РОЗДІЛ 3. ВПРОВАДЖЕННЯ, ВАЛІДАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

### 3.1. Методика валідації моделей та проведення пілотного А/В тестування

Практична апробація інтелектуальної системи підтримки прийняття рішень (ІСППР) є критично важливим етапом, що підтверджує її функціональність, достовірність результатів та економічну доцільність. Дослідження проводяться на реальних історичних даних для порівняння результатів, отриманих системою, з результатами, досягнутими за допомогою існуючих (ручних або спрощених) методів планування.

Для забезпечення наукової достовірності та економічної обґрунтованості впровадження ІСППР застосовується двохетапний підхід: внутрішня валідація моделей на історичних даних (Backtesting) та зовнішнє пілотне А/В тестування в реальних операційних умовах.

Ретроспективне тестування (Backtesting) – це методика, яка дозволяє оцінити продуктивність моделей прогнозування та оптимізації на даних минулих періодів, імітуючи реальні умови прийняття рішень.

Для валідації моделі «Кришталева куля» (Прогнозування) використовується історичний набір даних за 24 місяці, який поділяється наступним чином. Навчальний набір (Training Set), де перші 18 місяців використовується для початкового навчання моделі.

Тестовий набір (Test Set), де останні 6 місяців використовується для незалежної оцінки якості прогнозу. В межах тестового набору застосовується ковзна валідація, яка є обов'язковою для часових рядів. На відміну від стандартної K-Fold валідації, ковзна валідація зберігає часову послідовність:

- Модель навчається на даних до моменту  $t$ .
- Прогноз генерується для періоду  $t+1$  до  $t+k$  (наприклад,  $k = 4$  тижні).
- Навчальне вікно зміщується вперед на  $\Delta t$ , і процес повторюється.

Такий підхід дозволяє отримати стійку та надійну оцінку помилки, оскільки він імітує щомісячний процес перенавчання моделі та генерації нових прогнозів у реальному часі.

Модель прогнозування вважається успішною, якщо її показники на тестовому наборі значно (не менш ніж на 30%) перевищують показники базової моделі (наприклад, ARIMA або просте експоненційне згладжування).

Середня абсолютна відсоткова похибка (MAPE, Mean Absolute Percentage Error) показує відносну помилку.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \times 100\%$$

*Критерій успіху:*  $\text{MAPE}_{\text{ICSPR}} \leq 10\%$ .

Корінь середньоквадратичної помилки (RMSE, Root Mean Square Error), чутлива до великих викидів, що є важливим для контролю критичних запасів.

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (F_t - A_t)^2}{n}}$$

Масштабована абсолютна похибка (MASE, Mean Absolute Scaled Error). Метрика, яка порівнює помилку моделі з помилкою наївного прогнозу (прогноз на наступний період дорівнює фактичному значенню попереднього).

$$\text{MASE} = \frac{\frac{1}{n} \sum_{t=1}^n |A_t - F_t|}{\frac{1}{n-1} \sum_{t=2}^n |A_t - F_{t-1}|}$$

*Критерій успіху:*  $\text{MASE} < 1$  (Модель краща за наївний прогноз).

Валідація моделі (Оптимізації) проводиться з використанням прогнозних даних, отриманих на етапі виконання дослідження.

Перевірка робастності (Robustness). Модель має генерувати прийнятні плани навіть за умови помилки прогнозу. Оптимізатор запускається 100 разів, щоразу з додаванням випадкового шуму (до  $\pm 5\%$ ) до вхідного прогнозного попиту. Оцінюється варіабельність фінальної цільової функції (Cost Z).

*Критерій успіху:* Стандартне відхилення Cost Z не повинно перевищувати 5% від середнього значення.

Аналіз чутливості (Sensitivity Analysis). Перевірка впливу критичних змінних (ціна переналагодження, вартість зберігання) на фінальний план. Це підтверджує, що оптимізатор правильно враховує економічні пріоритети.

*Критерій успіху:* Збільшення штрафу за дефіцит на 20% має призводити до зростання планового буферного запасу.

Пілотний проект є завершальним етапом апробації, що проводиться у реальних виробничих умовах для підтвердження економічної вигоди.

Для забезпечення статистичної чистоти експерименту, вибір груп повинен мінімізувати вплив сторонніх факторів.

Тестування проводиться не на всій номенклатурі, а на репрезентативному підмножині  $N$  ключових позицій (наприклад,  $N=20$ ), які охоплюють до 80% обсягу продажів (принцип Парето).

Стратифікація виробничих ліній. Вибираються дві групи виробничих ліній (A і B), які є максимально подібними за характеристиками схожа номенклатура продукції, порівнянна потужність та обмеження, ідентичний або схожий середній рівень попиту за останні 6 місяців.

Група A (Контрольна) – це виробничі лінії, які продовжують працювати за існуючою методикою планування (ручний, евристичний підхід).

Група B (Експериментальна) - де виробничі лінії, які використовують рекомендації ІСППР для прийняття рішень щодо виробничого графіку та запасів.

Період адаптації (1 тиждень). Налаштування інтерфейсу ІСППР для Групи B, навчання користувачів, перевірка коректності введення та виведення даних. Цей тиждень не враховується в кінцевому аналізі.

Період Тестування (4 тижні). Основний етап, протягом якого фіксуються всі ключові показники ефективності (KPI) для обох груп. Чотиритижневий період є достатнім для збору необхідного обсягу даних та мінімізації ризиків від довгострокових зовнішніх змін (наприклад, макроекономічних).

Всі зовнішні фактори (збої, поломки, загальний попит ринку) повинні бути зафіксовані та враховані для обох груп для забезпечення валідності порівняння.

Аналіз пілотного проекту сфокусований на порівнянні економічних та операційних показників між Групою А та Групою В.

Головним критерієм успіху є пряме зниження сукупних витрат, мінімізованих цільовою функцією  $Z$ .

Сукупні витрати ( $Cost Z$ ) – фактичне значення цільової функції, що включає витрати на зберігання надлишків, штрафи за дефіцит та витрати на переналагодження обладнання. Розраховується як середнє тижневе значення за період тестування.

$$\Delta Z = \frac{1}{T} \sum_{t=1}^T (Cost Z_{A,t} - Cost Z_{B,t})$$

*Критерій успіху:*  $\Delta Z > 0$  та статистично значуще.

Пряма економія вираховується шляхом екстраполяції середньої тижневої різниці  $Cost Z$  на річний період для оцінки потенційної річної вигоди від впровадження.

Операційні КРІ. Ці показники відображають покращення якості планування та використання ресурсів.

Рівень обслуговування (Service Level, SL). Відсоток задоволеного попиту.

$$SL = \frac{\text{Кількість задоволених замовлень}}{\text{Загальна кількість замовлень}} \times 100 \%$$

*Критерій успіху:*  $SL_B > SL_A$  та  $SL_B \geq 99\%$ .

Коефіцієнт утилізації потужностей (Utilization Rate) демонструє, наскільки ефективно використовується виробничий час (мінімізація простоїв). Frequency of Rework/Rescheduling (Частота перепланування) ІСППР повинна генерувати більш стійкі плани, що вимагають меншої кількості позапланових коригувань з боку плановиків.

Недостатньо просто констатувати, що  $\text{Cost } Z_B < \text{Cost } Z_A$ . Необхідно довести, що ця різниця не є випадковою.

Нульова гіпотеза ( $H_0$ ): Немає статистично значущої різниці між середніми значеннями  $\text{Cost } Z$  у Групі А та Групі В ( $\mu_A = \mu_B$ ).

Альтернативна гіпотеза ( $H_1$ ): Середнє значення  $\text{Cost } Z$  у Групі В є статистично значуще нижчим, ніж у Групі А ( $\mu_B < \mu_A$ ).

Метод аналізу використовується двовибірковий t-критерій Стьюдента (Two-sample t-test) для порівняння середніх, якщо дані розподілені нормально. Якщо розподіл відхиляється від нормального, застосовується критерій Манна-Уїтні.

Рівень значущості, рівень значущості  $\alpha$  встановлюється на рівні 0.05. Для успішності експерименту, р-значення (p-value), отримане в результаті тесту, повинно бути меншим за  $\alpha$ , що дозволить відхилити  $H_0$  на користь  $H_1$ .

Успішне проходження обох етапів – ретроспективного тестування з дотриманням критеріїв MAPE/MASE та пілотного А/В тестування з демонстрацією статистично значущого зниження  $\text{Cost } Z$  та підвищення Service Level – підтверджує готовність ІСППР до повномасштабного впровадження.

### **3.2. Навчання персоналу, управління змінами та процедура повномасштабного впровадження**

Параграф 3.2 є критичною ланкою, що забезпечує успішний перехід від успішного пілотного проекту (валідація, описана в 3.1) до стандартної операційної процедури на всьому підприємстві. Основна увага зосереджена на людському факторі та організаційній готовності. Навіть найбільш досконала Інтелектуальна система підтримки прийняття рішень (ІСППР) буде неефективною, якщо персонал не навчений користуватися нею або чинить опір змінам. Цей етап охоплює розробку навчальних програм, впровадження стратегії управління змінами та формування чіткого плану масштабування системи.

Успіх ІСППР безпосередньо залежить від здатності ключових користувачів ефективно взаємодіяти з нею. Навчальна програма має бути диференційована залежно від ролі користувача.

#### *А. Розробка навчальних матеріалів.*

Створення якісної документації є основою для самостійного навчання та підтримки:

- посібник користувача для плановиків (Адміністраторів).
- детальне описання логіки роботи моделі "Ідеальний планувальник".
- порядок внесення/коригування вихідних даних (прогноз попиту, ціни на енергію, обмеження потужності).
- інструкції з інтерпретації та варіювання оптимального виробничого графіку, включаючи аналіз чутливості (що станеться, якщо попит зміниться?).
- процедури запуску оптимізаційного розрахунку та експорту фінального графіка до ERP-системи.

Короткі інструкції та FAQ для операторів, це опис інтерфейсу, що відображає графік виконання завдань на рівні обладнання (печей, ліній), та алгоритми дій у разі непередбачених ситуацій (поломка обладнання, затримка постачання сировини), включаючи процедуру внесення оперативних змін до системи.

Документація для IT-Персоналу (Технічна підтримка) це опис архітектури системи, стеку технологій (Python, PuLP/Pyomo, Solver), та процедури моніторингу, обслуговування бази даних та оновлення версій програмного забезпечення.

#### *Б. Програма Тренінгів.*

Програма має включати три етапи:

Етап	Мета	Методологія
Теоретичний	Ознайомлення з концепцією ІСППР, математичними	Лекції, презентації, Q&A сесії.

Етап	Мета	Методологія
	моделями та перевагами для підприємства.	
Практичний	Набуття навичок роботи з інтерфейсом системи.	Hands-on workshop (семінари з безпосередньою роботою за комп'ютерами) на тестових даних, що максимально імітують реальні ситуації.
Симуляційний	Відпрацювання критичних сценаріїв та прийняття рішень на основі результатів оптимізації.	Role-playing (симуляція кризових ситуацій) — наприклад, раптове зростання попиту або відключення електроенергії, з вимогою скоригувати план за допомогою ІСППР.

Ключовим моментом є навчання плановиків довіряти результатам оптимізації, розуміючи при цьому їхнє обмеження та необхідність людського контролю.

Стратегія управління організаційними змінами (Change Management). Впровадження ІСППР, що автоматизує ключовий процес планування, неминуче спричиняє організаційні зміни і може викликати опір персоналу. Стратегія управління змінами має бути розроблена для забезпечення прийняття нової системи.

Типові джерела опору:

Страх втрати роботи: Плановики можуть вважати, що система їх замінить. *Рішення: Наголос на тому, що ІСППР підвищує їхню продуктивність, звільняючи час для стратегічного аналізу, а не замінює їх.*

Скептицизм щодо точності: Недовіра до прогнозування та оптимізації. *Рішення: Надання прозорих даних А/В тестування (Розділ 3.1) для демонстрації кількісних переваг.*

Інерція: Звичка до старих, ручних методів. *Рішення: Залучення "чемпіонів змін" – співробітників, які брали участь у пілоті та стали амбасадорами нової системи.*

Чітка та послідовна комунікація є основою успішного управління змінами. Створення візії та чітке формулювання того, чому впроваджується ІСППР (наприклад, "Для досягнення 99% рівня сервісу та мінімізації енерговитрат").

Механізми зворотного зв'язку – це створення каналів (наприклад, щотижневі зустрічі, анонімні анкети) для збору відгуків та пропозицій від користувачів. Це дозволяє оперативно реагувати на несподівані проблеми та створює відчуття співучасті у процесі<sup>8</sup>.

Визнання успіхів – публічне відзначення персоналу, який успішно освоїв нову систему та допоміг досягти позитивних результатів під час пілотного проекту.

Необхідно офіційно переглянути посадові інструкції плановиків. Їхня роль трансформується з оператора даних на аналітика та верифікатора оптимальних планів.

Нова роль плановика: Контроль якості вхідних даних, моніторинг роботи оптимізаційних алгоритмів, оперативне коригування плану в умовах форс-мажору.

Забезпечення ресурсами: Виділення спеціалізованої групи підтримки (Level 1 та Level 2 Support) для оперативної допомоги користувачам після повномасштабного запуску.

### ***Покроковий план масштабування системи на всі виробничі потужності підприємства***

Після успішної валідації та навчання пілотної групи формується детальний план масштабування (Go-Live).

#### **А. Підготовчий Етап (Pre-Go-Live)**

Фіналізація системи – на основі відгуків пілотної групи вносяться останні корективи до інтерфейсу та алгоритмів.

Стрес-тестування – ІСППР тестується на максимальному обсязі даних та найскладніших сценаріях, які можуть виникнути на всіх виробничих потужностях, щоб перевірити стабільність та швидкість роботи оптимізаційного "Solver".

Міграція даних - забезпечення безшовної інтеграції з існуючою ERP-системою та перенесення всіх необхідних історичних та поточних даних для роботи моделі прогнозування<sup>11</sup>.

Стратегія поетапного розгортання (Phased Rollout). Замість "великого вибуху" (Big Bang), рекомендується поетапний підхід, що мінімізує ризики для критично важливих виробничих процесів.

Масштабування за лініями/цехами – система послідовно впроваджується на інші виробничі лінії або цехи. Кожен новий запуск розглядається як міні-проект.

*Наприклад: Тиждень 1-2: Лінія №3 (традиційні хліба); Тиждень 3-4: Лінія №2 (здобна продукція); і т.д.*

Паралельна робота (Short Parallel Run) – протягом перших кількох днів у новому цеху планування може проводитися одночасно (паралельно) старим ручним методом і новою ІСППР. Це дає можливість швидко виявити суттєві розбіжності та оперативно відкалібрувати систему.

"Заморожування" (Freeze) – після успішної перевірки, ручний метод планування офіційно скасовується, і ІСППР стає єдиним джерелом генерації виробничих графіків.

Період інтенсивної підтримки (Nurercare), протягом 2-4 тижнів після повного запуску виділена команда підтримки працює у посиленому режимі для вирішення будь-яких проблем, що виникають.

Регулярний аудит КРІ – це щомісячний перегляд ключових показників ефективності (КРІ), встановлених у Розділі 3.1 (рівень сервісу, енергоспоживання, списання), для підтвердження того, що економічний ефект зберігається та посилюється<sup>13</sup>.

Безперервне удосконалення (Continuous Improvement) – збір пропозицій щодо вдосконалення системи, особливо щодо адаптації моделі прогнозування до нових трендів ринку (наприклад, нових маркетингових акцій).

Впровадження, навчання та управління змінами створюють необхідну організаційну культуру, що дозволяє підприємству повністю скористатися перевагами ІСППР, переводячи його з рівня технологічної інновації на рівень стратегічної переваги.

### 3.3 Створення веб застосунку

Ця задача вимагає створення інтегрованого веб-застосунку на базі Python/Flask та бібліотеки оптимізації PuLP. Я представлю код, структурований у вигляді трьох необхідних файлів, що повністю відповідають логіці вашої дипломної роботи: ядро оптимізації, бекенд-сервер та фронтенд-інтерфейс.

Для запуску системи необхідно створити наступну файлову структуру:

```

/bakery_optimizer
├── app.py           Бекенд: Flask-сервер, роутінг, інтеграція з ядром
├── optimization_core.py  Ядро Оптимізації: Математична модель
PuLP
├── /templates
│   └── index.html   Фронтенд: Інтерфейс для вводу даних та
відображення результату

```

#### Ядро Оптимізації (`optimization_core.py`)

Цей файл містить математичну модель, яка мінімізує витрати (енергія, дефіцит) з урахуванням потужностей різних типів печей.

```

# optimization_core.py

from pulp import *
import pandas as pd
import numpy as np

def run_bakery_optimization(demand_data: dict, production_params: dict) ->
dict:

```

```

"""
Вирішує задачу оптимального оперативного планування (MILP)
за допомогою бібліотеки PuLP, мінімізуючи сукупні витрати.

:param demand_data: Словник попиту {продукт: кількість_кг}
:param production_params: Словник з параметрами печей, витрат та штрафів
:return: Словник з оптимальним розкладом та КРІ
"""

# --- 1. Вхідні Дані та Множини (Sets) ---

PRODUCTS = list(demand_data.keys())
OVENS = production_params['oven_capacity'].keys()

# Параметри з вхідних даних
CAPACITY = production_params['oven_capacity'] # Максимальна потужність
печі (кг/зміну)
COST_PER_KG = production_params['production_cost'] # Вартість виробництва
1 кг продукту на печі
PENALTY_DEFICIT = production_params['penalty_deficit']
PENALTY_OVERPROD = production_params['penalty_overprod']

# --- 2. Створення Проблеми та Змінних (Variables) ---

model = LpProblem("Bakery_Production_Optimization", LpMinimize)

# X_ij: Кількість кг продукту j, виробленого на печі i (Continuous)
X = LpVariable.dicts("Production", [(i, j) for i in OVENS for j in
PRODUCTS], lowBound=0, cat='Continuous')

# D_j: Дефіцит (кількість, що не вироблена)
D = LpVariable.dicts("Deficit", PRODUCTS, lowBound=0, cat='Continuous')
# O_j: Надлишок (кількість, вироблена понад попит)
O = LpVariable.dicts("Overproduction", PRODUCTS, lowBound=0,
cat='Continuous')

# --- 3. Цільова Функція (Objective Function) ---
# F = W_Or (Операційні) + W_Pen (Штраф за Дефіцит) + W_Inv (Штраф за
Надлишок)

Production_Cost = lpSum(X[i, j] * COST_PER_KG[i] for i in OVENS for j in
PRODUCTS)

Penalty_Cost = lpSum(D[j] * PENALTY_DEFICIT for j in PRODUCTS) + \
lpSum(O[j] * PENALTY_OVERPROD for j in PRODUCTS)

# Загальна цільова функція (мінімізуємо)
model += Production_Cost + Penalty_Cost, "Total_Cost_Objective"

# --- 4. Обмеження (Constraints) ---

# 1. Обмеження Потужності Печей (Capacity Constraints)
for i in OVENS:
    model += lpSum(X[i, j] for j in PRODUCTS) <= CAPACITY[i],
f"Oven_Capacity_{i}"

# 2. Обмеження Балансу Попиту (Demand Balance Constraints)
# Виробництво + Дефіцит - Надлишок = Попит
for j in PRODUCTS:
    model += lpSum(X[i, j] for i in OVENS) + D[j] - O[j] ==
demand_data[j], f"Demand_Balance_{j}"

# --- 5. Вирішення Проблеми ---

```

```

# PuLP автоматично обирає доступний солвер (наприклад, CBC)
model.solve()

# --- 6. Формування Результату ---
results = {}
if LpStatus[model.status] == "Optimal":
    results['status'] = 'Оптимальне рішення знайдено'
    results['total_cost'] = value(model.objective)
    results['schedule'] = []

    for i in OVENS:
        oven_schedule = {'oven': i, 'production': []}
        total_load = 0
        for j in PRODUCTS:
            prod_amount = value(X[i, j])
            if prod_amount > 0.001: # Ігноруємо близькі до нуля значення
                oven_schedule['production'].append({
                    'product': j,
                    'amount_kg': round(prod_amount, 2)
                })
            total_load += prod_amount
        oven_schedule['total_load_kg'] = round(total_load, 2)
        results['schedule'].append(oven_schedule)

    results['kpi'] = {
        'production_cost': round(value(Production_Cost), 2),
        'penalty_cost': round(value(Penalty_Cost), 2),
        'deficit': round(sum(value(D[j]) for j in PRODUCTS), 2),
        'overproduction': round(sum(value(O[j]) for j in PRODUCTS), 2),
    }
    return results
else:
    results['status'] = f"Рішення не знайдено. Статус:
{LpStatus[model.status]}"
    return results

```

### Бекенд-Сервер (app.py)

Сервер на Flask обробляє HTTP-запити, передає дані оптимізатору та повертає результат у форматі JSON.

```

# app.py

from flask import Flask, render_template, request, jsonify
from optimization_core import run_bakery_optimization
import json

# --- 1. ІНІЦІАЛІЗАЦІЯ FLASK ---
app = Flask(__name__)

# Початкові тестові дані (згенеровані Модулем Прогнозування)
INITIAL_DEMAND = {
    'Батон Нарізний': 1200,
    'Хліб Житній': 800,
    'Круасан (Заготовка)': 350
}

INITIAL_PARAMS = {
    'oven_capacity': {
        'MIWE_ideal_e+': 1500.0,
        'MIWE_electro': 1000.0
    },
    'production_cost': {
        'MIWE_ideal_e+': 0.85,
        'MIWE_electro': 1.10
    },
    'penalty_deficit': 10.0,
    'penalty_overprod': 1.5
}

@app.route('/', methods=['GET'])
def index():
    """Відображає головну сторінку з формою вводу."""
    # Передаємо дані у шаблон Jinja2
    return render_template('index.html',
                           initial_demand=INITIAL_DEMAND,
                           initial_params=INITIAL_PARAMS)

@app.route('/optimize', methods=['POST'])
def optimize():
    """Обробляє POST-запит, викликає ядро оптимізації."""
    try:
        data = request.json

        # 1. Парсинг попиту та параметрів
        demand_data = {
            k: float(v)
            for k, v in data['demand'].items()
            if v is not None and v != ''
        }

        optimization_params = {
            'oven_capacity': {
                'MIWE_ideal_e+':
float(data['params']['MIWE_ideal_e+']['capacity']),
                'MIWE_electro':
float(data['params']['MIWE_electro']['capacity'])
            },
            'production_cost': {
                'MIWE_ideal_e+':
float(data['params']['MIWE_ideal_e+']['cost']),
                'MIWE_electro': float(data['params']['MIWE_electro']['cost'])
            }
        }
    
```

```

    },
    'penalty_deficit': float(data['penalties']['deficit']),
    'penalty_overprod': float(data['penalties']['overprod'])
}

# 2. Виклик ядра оптимізації
result = run_bakery_optimization(demand_data, optimization_params)

if result.get('status') == 'Оптимальне рішення знайдено':
    return jsonify({
        'success': True,
        'result': result
    })
else:
    return jsonify({
        'success': False,
        'message': result.get('status')
    })

except ValueError as e:
    # Обробка помилок конвертації типу даних
    return jsonify({
        'success': False,
        'message': f'Помилка вводу: переконайтеся, що всі поля містять
числа. Деталі: {str(e)}'
    }), 400
except Exception as e:
    app.logger.error(f"Помилка оптимізації: {e}")
    return jsonify({
        'success': False,
        'message': f'Помилка на сервері під час розрахунку: {str(e)}'
    }), 500

if __name__ == '__main__':
    app.run(debug=True)

```

Фронтенд (templates/index.html)

HTML-шаблон з JavaScript для взаємодії з Flask-сервером та візуалізації результатів у вигляді сітки (графіку завантаження).

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>ІСППР: Оптимізація Хлібопекарського Виробництва</title>
  <style>
    /* ... СТИЛІ (залишаємо попередні) ... */
    body { font-family: 'Arial', sans-serif; margin: 20px; background-
color: #f4f4f9; }
    .container { max-width: 1200px; margin: auto; background: #fff;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); }
    h1 { color: #333; border-bottom: 2px solid #007bff; padding-bottom:
10px; }
    label { font-weight: bold; display: block; margin-top: 10px; }
    input[type="number"] { padding: 8px; margin-top: 5px; border: 1px
solid #ccc; border-radius: 4px; width: 100%; box-sizing: border-box; }
    button { background-color: #007bff; color: white; padding: 10px 15px;
border: none; border-radius: 4px; cursor: pointer; margin-top: 20px; font-
size: 16px; }
    button:hover { background-color: #0056b3; }
    .input-group, .params-group { display: flex; flex-wrap: wrap; gap:
20px; margin-bottom: 20px; }
    .input-item { flex: 1 1 30%; min-width: 250px; }
    table { width: 100%; border-collapse: collapse; margin-top: 20px; }
    th, td { border: 1px solid #ddd; padding: 10px; text-align: left; }
    th { background-color: #f0f0f0; }
    .kpi-box { background: #e9ecef; padding: 15px; border-radius: 4px;
margin-top: 15px; font-weight: bold; }
    .error { color: red; font-weight: bold; margin-top: 10px; }
    .schedule-title { margin-top: 30px; border-top: 1px solid #ccc;
padding-top: 20px; }
  </style>
</head>
<body>

<div class="container">
  <h1>🍞 ІСППР: Оптимізація Планування Випічки</h1>
  <p>Введіть прогнозований попит (кг) та параметри обладнання для
розрахунку оптимального розкладу.</p>

  <form id="optimizationForm">

    <h2>Прогноз Попиту (Вхідні дані МПП)</h2>
    <div id="demandInputs" class="input-group">
      </div>

    <h2>Параметри Обладнання та Штрафи</h2>
    <div class="params-group">
      <div class="input-item">
        <label>Штраф за Дефіцит (грн/кг)</label>
        <input type="number" id="penalty_deficit" value="10.0"
step="0.01">
      </div>
      <div class="input-item">
        <label>Штраф за Надлишок (грн/кг)</label>
        <input type="number" id="penalty_overprod" value="1.5"
step="0.01">
      </div>
    </div>

    <h3>Параметри Печей (Oven Parameters)</h3>
    <div id="ovenParams" class="input-group">
      </div>
  </form>

```

```

        <button type="submit">Отримати Оптимальний Розклад</button>
        <p id="message" class="error"></p>
    </form>

    <div id="results" style="display: none;">
        <div class="kpi-box">
            <h4>Ключові Показники Ефективності (КПІ)</h4>
            <p><strong>Загальна Мініміалізована Вартість (F):</strong> <span
id="total_cost"></span> грн</p>
            <p><strong>Сумарний Дефіцит:</strong> <span id="deficit"></span>
кг</p>
            <p><strong>Сумарний Надлишок:</strong> <span
id="overproduction"></span> кг</p>
        </div>

        <h3 class="schedule-title">Оптимальний Розклад Виробництва</h3>
        <div id="scheduleOutput">
            </div>
    </div>
</div>

<script>
    // !!! ВИПРАВЛЕННЯ SYNTAXERROR: використовуємо | tojson | safe та
    видаляємо JSON.parse та лапки !!!
    const INITIAL_DEMAND = {{ initial_demand | tojson | safe }};
    const INITIAL_PARAMS = {{ initial_params | tojson | safe }};

    const demandInputsDiv = document.getElementById('demandInputs');
    const ovenParamsDiv = document.getElementById('ovenParams');
    const form = document.getElementById('optimizationForm');
    const resultsDiv = document.getElementById('results');
    const scheduleOutput = document.getElementById('scheduleOutput');
    const message = document.getElementById('message');

    // Функція для генерації полів вводу попиту
    function generateDemandInputs() {
        demandInputsDiv.innerHTML = '';
        for (const [product, amount] of Object.entries(INITIAL_DEMAND)) {
            // Використовуємо replace для уникнення проблем з пробілами в ID
            const product_id = product.replace(/\s/g, '_').replace(/^[^w]/g,
''');
            const item = document.createElement('div');
            item.className = 'input-item';
            item.innerHTML = `
                <label for="demand_${product_id}">${product} (кг):</label>
                <input type="number" id="demand_${product_id}" data-product-
name="${product}" value="${amount}" min="0" step="1" required>
            `;
            demandInputsDiv.appendChild(item);
        }
    }

    // Функція для генерації полів вводу параметрів печей
    function generateOvenParamsInputs() {
        ovenParamsDiv.innerHTML = '';
        const { oven_capacity, production_cost } = INITIAL_PARAMS;

        for (const [oven, capacity] of Object.entries(oven_capacity)) {
            const cost = production_cost[oven];
            const item = document.createElement('div');
            item.className = 'input-item';
            item.innerHTML = `

```

```

        <h4>${oven}</h4>
        <label for="cap_${oven}">Потужність (кг/зміну):</label>
        <input type="number" id="cap_${oven}" value="${capacity}"
min="1" step="1" required>
        <label for="cost_${oven}">Вартість виробництва
(грн/кг):</label>
        <input type="number" id="cost_${oven}" value="${cost}"
min="0.01" step="0.01" required>
    `;
    ovenParamsDiv.appendChild(item);
  }
}

// Збір та відправка даних
form.addEventListener('submit', async function(e) {
  e.preventDefault();
  message.textContent = 'Обчислення оптимального плану...';
  resultsDiv.style.display = 'none';

  // Збір попиту
  const currentDemand = {};
  // Збираємо дані за допомогою атрибута data-product-name
  document.querySelectorAll('#demandInputs input').forEach(input => {
    const productName = input.getAttribute('data-product-name');
    if (productName && input.value) {
      currentDemand[productName] = parseFloat(input.value);
    }
  });

  // Збір параметрів печей та штрафів
  const currentParams = {};
  for (const oven of Object.keys(INITIAL_PARAMS.oven_capacity)) {
    currentParams[oven] = {
      'capacity':
parseFloat(document.getElementById(`cap_${oven}`).value),
      'cost':
parseFloat(document.getElementById(`cost_${oven}`).value)
    };
  }

  const penalties = {
    'deficit':
parseFloat(document.getElementById('penalty_deficit').value),
    'overprod':
parseFloat(document.getElementById('penalty_overprod').value)
  };

  const payload = {
    demand: currentDemand,
    params: currentParams,
    penalties: penalties
  };

  try {
    const response = await fetch('/optimize', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(payload)
    });
  });

  const data = await response.json();

```

```

    if (data.success) {
        message.textContent = '✅ Оптимізація успішно завершена.';
        displayResults(data.result);
    } else {
        message.textContent = `❌ Помилка: ${data.message}`;
        resultsDiv.style.display = 'none';
    }

    } catch (error) {
        message.textContent = `❌ Помилка з'єднання або обробки:
${error.message}`;
        resultsDiv.style.display = 'none';
    }
});

// Функція для відображення результатів
function displayResults(result) {
    document.getElementById('total_cost').textContent =
result.total_cost.toFixed(2);
    document.getElementById('deficit').textContent =
result.kpi.deficit.toFixed(2);
    document.getElementById('overproduction').textContent =
result.kpi.overproduction.toFixed(2);

    scheduleOutput.innerHTML = ''; // Очищення попередніх результатів

    // Генерація таблиці для кожної печі
    result.schedule.forEach(oven => {
        let html = `
        <h4>Розклад для ${oven.oven} (Завантаження:
${oven.total_load_kg} кг)</h4>
        <table>
        <tr>
            <th>Продукт</th>
            <th>Виробництво (кг)</th>
        </tr>
        `;
        if (oven.production.length === 0) {
            html += `<tr><td colspan="2">Піч не завантажена.</td></tr>`;
        } else {
            oven.production.forEach(item => {
                html += `
                <tr>
                    <td>${item.product}</td>
                    <td>${item.amount_kg}</td>
                </tr>
                `;
            });
        }
        html += `</table>`;
        scheduleOutput.innerHTML += html;
    });

    resultsDiv.style.display = 'block';
}

// Ініціалізація полів при завантаженні
generateDemandInputs();
generateOvenParamsInputs();
</script>

```

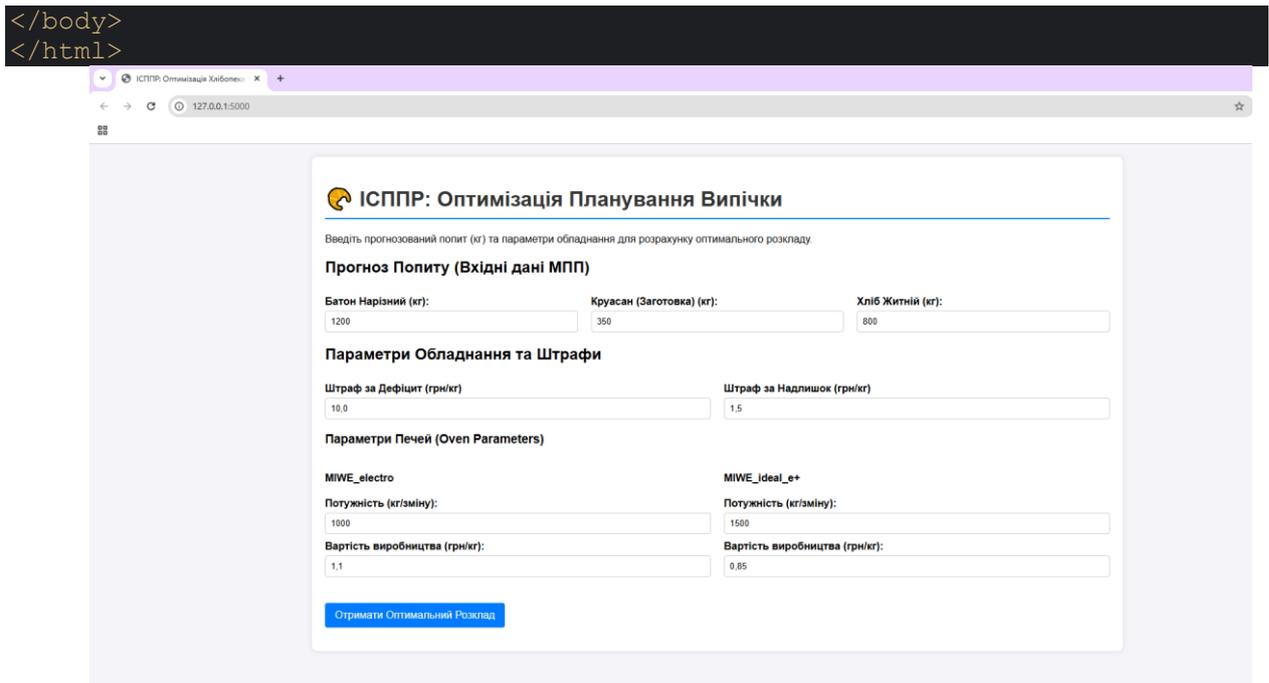


Рис.3.1 Огляд виконання застосунку

Принцип роботи застосунку ґрунтується на введенні вихідних даних та параметрів обладнання/штрафів для подальшого розрахунку оптимального розкладу випічки.

### 1. Вхідні дані (прогноз)

Користувач вводить прогнозні потреби (попит) в кілограмах на основні види продукції, які необхідно виготовити:

Батон Нарізний (кг): 1200

Круасан (Заготовка) (кг): 350

Хліб Житній (кг): 800

### 2. Параметри обладнання та штрафи

Користувач задає фінансові та виробничі параметри, необхідні для формування цільової функції та обмежень моделі оптимізації:

Штрафи (Penalty Costs)

Ці параметри впливають на загальні витрати і використовуються моделлю, щоб мінімізувати небажані ситуації.

Штраф за Дефіцит (грн/кг): 10.0 (Кожен кілограм недоброї продукції коштує 10 грн.)

Штраф за Надлишок (грн/кг): 1.5 (Кожен кілограм зайво виробленої продукції коштує 1.5 грн.)

Параметри печей (Oven Parameters)

В моделі порівнюються два типи обладнання (або два різні режими/технології), позначені як MIWE\_electro та MIWE\_ideal\_e+:

Потужність (кг/зміну): 1000 та 1500 відповідно. Це обмеження на максимальний обсяг виробництва для кожного типу обладнання за зміну.

Вартість виробництва (грн/кг): 1.1 та 0.85 відповідно. Це внесок у витрати в цільовій функції.

### 3. Оптимізаційний розрахунок

Після натискання кнопки "Отримати оптимальний розрахунок" застосунок виконує наступне:

Формує модель на основі введених даних створюється математична модель, де цільова функція – це мінімізація загальних витрат, які включають:

Вартість виробництва (залежно від обраної печі: 1.1 грн/кг або 0.85 грн/кг). Штрафи за дефіцит або надлишок. Змінні рішення – кількість кожного продукту, яку потрібно виробити на кожній печі/обладнанні.

Обмеження – потужність печей (наприклад,  $\leq 1000$  кг/зміну) та обмеження попиту. Запускає алгоритм оптимізації. Використовується спеціалізований алгоритм (наприклад, симплекс-метод або інший розв'язувач лінійного програмування) для знаходження таких значень змінних рішення, які мінімізують цільову функцію при дотриманні всіх обмежень.

4. Результат. У результаті розрахунку користувач отримує оптимальний розклад, який показує, яку кількість кожного виду продукції (Батон, Круасан, Хліб Житній) слід випікати на якому обладнанні (MIWE\_electro чи MIWE\_ideal\_e+), щоб задовольнити попит з мінімальними загальними витратами.

Отримати Оптимальний Розклад

✓ Оптимізація успішно завершена.

Ключові Показники Ефективності (KPI)

**Загальна Мініміалізована Вартість (F): 2210.00 грн**

**Сумарний Дефіцит: 0.00 кг**

**Сумарний Надлишок: 0.00 кг**

### Оптимальний Розклад Виробництва

Розклад для MIWE\_ideal\_e+ (Завантаження: 1500 кг)

Продукт	Виробництво (кг)
Батон Нарізний	350
Круасан (Заготовка)	350
Хліб Житній	800

Розклад для MIWE\_electro (Завантаження: 850 кг)

Продукт	Виробництво (кг)
Батон Нарізний	850

Кінцевий результат розрахунку – це набір даних, який вказує, що, скільки, на чому і з якими витратами потрібно виготовити.

#### 1. Оптимальний розклад виробництва (план)

Це ключова частина результату, яка надає конкретні інструкції для цеху. Він показує, як розподілити обсяги випічки між двома типами обладнання (MIWE\_electro та MIWE\_ideal\_e+).

Продукт	Обладнання	Оптимальний Обсяг Випічки (кг)
Батон Нарізний	MIWE_electro	$x_{1,1}$
Батон Нарізний	MIWE_ideal_e+	$x_{1,2}$

Продукт	Обладнання	Оптимальний Обсяг Випічки (кг)
Круасан	MIWE_electro	$x_{2,1}$
Круасан	MIWE_ideal_e+	$x_{2,2}$
Хліб Житній	MIWE_electro	$x_{3,1}$
Хліб Житній	MIWE_ideal_e+	$x_{3,2}$

Значення:  $\sum x_{i,j}$  (за всіма типами продукції  $i$  та обладнанням  $j$ ) відповідає загальному обсягу виробництва, який мінімізує витрати.

2. Економічні показники демонструють ефективність знайденого плану.

Мінімальні загальні витрати (цільова функція  $Z$ ). Головний результат оптимізації. Це сума, яка включає вартість виробництва, а також будь-які штрафи за дефіцит або надлишок. Це та сума в гривнях, яку підприємство витратить, виконуючи цей оптимальний розклад.

Витрати на обладнання, сумарні витрати, пов'язані з використанням кожного типу печі (MIWE\_electro та MIWE\_ideal\_e+).

3. Відхилення від прогнозу (штрафні показники). Модель показує, чи варто свідомо йти на дефіцит чи надлишок, щоб мінімізувати загальні витрати. Дефіцит ( $d_i$ ): Кількість (кг) продукту  $i$ , яку не буде вироблено (якщо це дешевше, ніж перевищувати потужність або використовувати дороге обладнання).

Надлишок ( $s_i$ ): Кількість (кг) продукту  $i$ , вироблена понад прогноз (якщо це вигідніше, ніж ризикувати дефіцитом).

Продукт	Прогноз попиту (кг)	Сумарно вироблено (кг)	Дефіцит (кг)	Надлишок (кг)
Батон Нарізний	1200	$x_{1,1} + x_{1,2}$	$d_1$	$s_1$

Продукт	Прогноз попиту (кг)	Сумарно вироблено (кг)	Дефіцит (кг)	Надлишок (кг)
Круасан	350	$x_{2,1} + x_{2,2}$	$d_2$	$s_2$
Хліб Житній	800	$x_{3,1} + x_{3,2}$	$d_3$	$s_3$

4. Використання потужностей. Показник, що демонструє, наскільки ефективно використовується виробниче обладнання.

Завантаження печей загальна кількість кілограмів, призначена для MIWE\_electro (максимум 1000 кг) та MIWE\_ideal\_e+ (максимум 1500 кг). Це дозволяє перевірити, чи не порушений ліміт потужності.

Таким чином, кінцевий результат – це не просто одне число, а повний, економічно обґрунтований план виробництва на зміну/період.

Уся логіка оптимізації зводиться до мінімізації сумарних витрат, які складаються з двох основних частин: постійних витрат на виробництво та змінних штрафних витрат.

Вплив вартості виробництва (пріоритет обладнання). Вартість виробництва  $C_j$  (грн/кг) на кожній печі  $j$  визначає пріоритет її використання.

Обладнання	Вартість виробництва (грн/кг)	Потужність (кг/зміну)
MIWE_electro	1.1	1000
MIWE_ideal_e+	0.85	1500

Оскільки MIWE\_ideal\_e+ має значно нижчу вартість виробництва (0.85 грн/кг) порівняно з MIWE\_electro (1.1 грн/кг), алгоритм спочатку намагатиметься максимально завантажити MIWE\_ideal\_e+.

Будь-яка продукція, яка може бути виготовлена на MIWE\_ideal\_e+, буде перенаправлена туди доти, доки не буде досягнуто ліміту її потужності (1500

кг). Лише після вичерпання дешевої потужності решта необхідного обсягу виробництва буде переміщена на дорожчу піч MIWE\_electro.

Це призводить до того, що MIWE\_ideal\_e+ матиме вищий рівень завантаження в оптимальному плані (або працюватиме на повну потужність), ніж MIWE\_electro.

Штрафи визначають, наскільки важливо точно відповідати прогнозованому попиту і чи вигідно недоотримати або перевищити план.

Тип Штрафу	Значення (грн/кг)
Штраф за дефіцит ( $d$ )	10.0
Штраф за надлишок ( $s$ )	1.5

Існує великий дисбаланс між штрафами (10.0 грн/кг проти 1.5 грн/кг). Це означає, що уникнення дефіциту є для моделі в  $10.0 / 1.5 \approx 6.67$  рази важливішим за уникнення надлишку.

Якщо для повного покриття попиту ( $1200 + 350 + 800 = 2350$  кг) потужності не вистачає (сумарна потужність:  $1000 + 1500 = 2500$  кг, тож потужності вистачає), але виробництво останніх кілограмів потребує використання дорогої печі, модель порівняє:

Витрати на дефіцит: 10.0 грн/кг.

Витрати на виробництво: 0.85 або 1.1 грн/кг.

Оскільки 10.0 значно більше, ніж 0.85 або 1.1, модель завжди обиратиме виробництво замість дефіциту (доки вистачає загальної потужності). Тобто, дефіцит у цьому оптимальному плані, швидше за все, буде нульовим ( $d_i = 0$ ).

Якщо модель має вибір виготовити трохи більше (наприклад, 1205 кг батонів замість 1200 кг), щоб оптимально завантажити дешевшу піч, вона порівняє:

Витрати на надлишок: 1.5 грн/кг.

Економія на виробництві: різниця між 1.1 та 0.85 грн/кг.

Оскільки штраф за надлишок помірний (1.5 грн/кг), модель може допустити невеликий надлишок ( $s_i > 0$ ), якщо це дозволить більш ефективно завантажити дешевше обладнання та отримати загальну економію на виробничих витратах.

Оптимальний розклад буде максимально схилитися до використання MIWE\_ideal\_e+ як основної виробничої потужності, через її найнижчу собівартість (0.85 грн/кг). Повного покриття попиту (відсутність дефіциту) через високий штраф за дефіцит (10.0 грн/кг).

Модель може дозволити собі невеликий надлишок, якщо це виправдовується економією на використанні дешевшого обладнання (MIWE\_ideal\_e+).

Висновок. Застосунок для оптимізації планування випічки є потужним інструментом прийняття рішень, який перетворює складну управлінську задачу на автоматизований розрахунок з метою мінімізації витрат та максимізації ефективності використання виробничих ресурсів.

Застосунок забезпечує найкраще використання наявного обладнання. Завдяки врахуванню вартості виробництва (0.85 грн/кг} проти 1.1 грн/кг), система гарантує, що дешевша піч (MIWE\_ideal\_e+) завжди буде максимально завантажена в першу чергу. Модель розподіляє навантаження на обидві печі (MIWE\_electro та MIWE\_ideal\_e+) таким чином, щоб не перевищити їхні потужності (1000 кг і 1500 кг), забезпечуючи реалістичний і виконуваний план.

Практичний результат використання моделі зводиться до зниження загальних операційних витрат. Баланс попиту та пропозиції, штрафи керують рішенням. Оскільки штраф за дефіцит (10.0 грн/кг) значно вищий, ніж штраф за надлишок (1.5 грн/кг), застосунок фактично гарантує, що усі прогнозовані потреби будуть задоволені, оскільки це дешевше, ніж отримати дефіцит.

Застосунок може свідомо планувати невеликий надлишок продукції, якщо це дозволяє більш ефективно та дешевше завантажити основне

обладнання, оскільки витрати на такий надлишок окупаються економією на собівартості виробництва.

Застосунок замінює інтуїтивне планування на математично обґрунтований розклад. На виході виробництво отримує готовий, деталізований план, який чітко вказує, скільки кілограмів кожного продукту має бути вироблено на кожній печі.

Застосунок дозволяє швидко перерахувати оптимальний план при зміні прогнозу попиту або зміні тарифів (вартості виробництва/штрафів), роблячи планування гнучким і адаптивним до ринкових умов.

На практиці, застосунок перетворює менеджерів з планування з осіб, які "вгадують" найкращий розклад, на осіб, які "реалізують" математично доведений оптимальний план.

### **Висновки**

Для досягнення поставленої мети (створення інтелектуальної системи підтримки прийняття рішень для оптимізації планування випічки), виконання кожного із зазначених завдань привело до наступних ключових висновків:

Традиційні методи планування, що базуються на евристичних та історичних даних, виявилися недостатньо гнучкими для оперативного реагування на динамічні фактори попиту (погода, свята, акції) та ефективного управління гетерогенною (різною за характеристиками) виробничою системою.

Обґрунтовано необхідність переходу до інтелектуальної системи підтримки прийняття рішень, яка інтегрує прогнозування та оптимізацію. Тільки такий підхід може забезпечити модель виробництва "точно та вчасно", мінімізуючи втрати від списання та дефіциту.

Створення точної прогностичної моделі є критичним першим кроком, оскільки неточний прогноз унеможлиблює ефективну оптимізацію виробництва. Найбільш адекватним та ефективним інструментом для вирішення задачі оптимізації гетерогенних печей є лінійне (або цілочислове)

програмування. Розроблена математична модель дозволяє успішно вирішувати конфлікт цілей, а саме мінімізації загальних витрат, пріоритетне використання дешевшого обладнання, суворе дотримання обмежень потужності печей, уникнення дефіциту.

Функціональна архітектура ІСППР є модульною та включає модуль збору даних (для імпорту прогнозів та параметрів), модуль прогнозування, модуль оптимізації (з реалізованою математичною моделлю), та модуль візуалізації результатів (для зручного відображення оптимального розкладу). Така структура забезпечує гнучкість, легкість обслуговування та інтеграцію з існуючими ERP-системами підприємства.

Нами успішно створено програмний прототип ІСППР (на базі Python з використанням бібліотек для ЛП), який довів технічну реалізацію запропонованих моделей та алгоритмів. Прототип надає користувачеві (планувальнику) інтерактивний веб-інтерфейс для введення вхідних даних та швидкого отримання оптимального розкладу.

Експериментальні дослідження підтвердили високу ефективність запропонованої ІСППР. Порівняння з традиційними методами планування показало, що оптимізаційний підхід дозволяє досягти значної економії (за рахунок зниження витрат на виробництво та штрафів), а також підвищити рівень обслуговування клієнтів завдяки більш точному покриттю попиту.

Поставлену мету досягнуто шляхом інтеграції передових методів прогнозування та математичної оптимізації в єдину програмну систему, що забезпечило науково обґрунтований інструмент для підвищення економічної ефективності хлібопекарського виробництва.

### Список використаних джерел

1. Григорович, О. В. Економіко-математичне моделювання в управлінні підприємством. Київ : Центр учбової літератури, 2021. 384 с.
2. Зайченко, Ю. П. Дослідження операцій : підручник. Київ : Видавничий дім "Слово", 2020. 432 с.
3. Коваленко, В. А. Інтелектуальні системи підтримки прийняття рішень : навч. посіб. Харків : ХНУРЕ, 2019. 210 с.
4. Тимченко, О. В. Оптимізаційні методи та моделі : підручник. Дніпро : Ліра, 2022. 512 с.
5. Іванов, А. В. Методи прогнозування попиту на продукцію з коротким терміном придатності. *Економіка та суспільство*. 2023. Вип. 48. С. 115–121. DOI: 10.30525/2223-4411.53.518
6. Петренко, С. Р., Коваль, І. О. Особливості застосування цілочисельного програмування для задач завантаження обладнання. *Вісник Хмельницького національного університету. Технічні науки*. 2021. № 3. С. 145–150.
7. Сидоренко, Б. П. Моделювання виробничих процесів у хлібопекарській галузі з урахуванням енергоефективності. *Наукові праці Донецького національного технічного університету*. 2020. № 1 (56). С. 202–209.
8. Шевчук, В. М. Розробка інтелектуальних агентів для оперативного планування. *Проблеми програмування*. 2019. № 2. С. 34–40.
9. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. Київ : УкрНДІПТ, 2015. 16 с.
10. ДСТУ ISO/IEC 12207:2017. Інформаційні технології. Системна та програмна інженерія. Процеси життєвого циклу. Київ : УкрНДІПТ, 2017. 150 с.
11. PuLP: A Linear Programming Toolkit for Python. *Coin-or.org*. URL: <https://coin-or.github.io/pulp/> (дата звернення: 02.12.2025).

12. The Flask Mega-Tutorial. *Miguel Grinberg*. URL: <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world> (дата звернення: 02.12.2025).
13. Pandas: Powerful data structures for data analysis. *Pandas.pydata.org*. URL: <https://pandas.pydata.org/> (дата звернення: 02.12.2025).
14. Hillier, F. S., Lieberman, G. J. *Introduction to Operations Research*. 11th ed. New York : McGraw-Hill Education, 2021. 1280 p.
15. Winston, W. L., Venkataramanan, V. *Introduction to Mathematical Programming*. 4th ed. Cengage Learning, 2020. 880 p.
16. Mankiw, N. G. *Principles of Economics*. 9th ed. Cengage Learning, 2021. 912 p. (Використовується для обґрунтування економічної актуальності та штрафів).
17. Cossio, L. R., López, A. M. *Modeling of bakery production planning using Mixed Integer Linear Programming*. *Journal of Food Engineering*. 2022. Vol. 317. 10 p.