

**Міністерство освіти і науки України
Тернопільський національний педагогічний університет
імені Володимира Гнатюка**

**Фізико-математичний факультет
Кафедра інформатики та методики її навчання**

**Кваліфікаційна робота
КОМП'ЮТЕРНИЙ ЕКСПЕРИМЕНТ В ЗАДАЧАХ ОПТИМІЗАЦІЇ ПРИ
ВИВЧЕННІ МОДЕЛЮВАННЯ В ЗАКЛАДАХ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ
ОСВІТИ**

**Спеціальність 014 Середня освіта
Освітня програма «Інформатика, математика, STEM-освіта»**

Здобувача другого (магістерського)
рівня вищої освіти
Куцого Сергія Володимировича

НАУКОВИЙ КЕРІВНИК:
кандидат фізико-математичних наук,
доцент Грод Інна Миколаївна

РЕЦЕНЗЕНТ:
завідувач кафедрою комп'ютерних
технологій Тернопільського
національного педагогічного
університету імені Володимира
Гнатюка,
кандидат технічних наук, доцент
Франко Юрій Павлович

Тернопіль – 2025

АНОТАЦІЯ

Куций С.В. Комп'ютерний експеримент в задачах оптимізації при вивченні моделювання в закладах загальної середньої освіти. Кваліфікаційна робота на здобуття освітнього ступеня «магістр» зі спеціальності 014.09 Середня освіта (Інформатика). ТНПУ ім. В. Гнатюка. Тернопіль, 2025.

Дане дослідження присвячене актуальній проблемі інтеграції методів математичного моделювання та оптимізації в освітній процес STEM-орієнтованої школи. У роботі розкрито сутність і класифікацію моделей, обґрунтовано дидактичний потенціал методів оптимізації та роль комп'ютерного експерименту в навчанні. Проаналізовано програмні засоби, придатні для використання у закладах загальної середньої освіти, та запропоновано методичні рекомендації, які можуть стати основою для створення інтегрованих курсів і факультативів.

Ключові слова: STEM-освіта, математичне моделювання, методи оптимізації, комп'ютерний експеримент, інтегроване навчання.

ABSTRACT

Kutsyi S.V. Computer experimentation in optimization problems within the study of modeling in secondary education establishments. Master's thesis for the degree of Master in 014.09 Secondary Education (Informatics). Ternopil Volodymyr Hnatiuk National Pedagogical University. Ternopil, 2025.

This study is devoted to the current problem of integrating mathematical modeling and optimization methods into the educational process of a STEM-oriented school. The paper reveals the essence and classification of models, substantiates the didactic potential of optimization methods and the role of computer experiments in education. Software tools suitable for use in general secondary education institutions are analyzed, and methodological recommendations are proposed that can become the basis for creating integrated courses and electives.

Keywords: STEM education, mathematical modeling, optimization methods, computer experiment, integrated learning.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ КОМП'ЮТЕРНОГО ЕКСПЕРИМЕНТУ В ЗАДАЧАХ ОПТИМІЗАЦІЇ.....	8
1.1. Поняття та класифікація математичного моделювання.....	8
1.2. Основи оптимізації та їх застосування в освітньому процесі.....	13
1.3. Роль комп'ютерного експерименту в навчанні моделюванню.....	18
1.4. Аналіз програмних засобів для проведення комп'ютерних експериментів...	21
ВИСНОВОК ДО РОЗДІЛУ 1.....	27
РОЗДІЛ 2. КОМП'ЮТЕРНИЙ ЕКСПЕРИМЕНТ В ЗАДАЧАХ ОПТИМІЗАЦІЇ	30
2.1. Розробка та апробація навчальних завдань з оптимізації.....	30
2.2. Використання комп'ютерного експерименту на уроках інформатики та математики на прикладі інтерактивного застосунку “School Optimization Experiment”.....	40
2.3. Методика впровадження задач оптимізації у процес навчання моделюванню.....	51
ВИСНОВОК ДО РОЗДІЛУ 2.....	60
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ.....	69

ВСТУП

Сучасний етап розвитку науки й техніки визначається зростанням обсягів даних, ускладненням технологічних процесів і підвищенням вимог до якості управлінських та інженерних рішень. У таких умовах інформаційні технології та математичні методи перестають бути допоміжними інструментами й перетворюються на базову основу практичної діяльності у виробництві, економіці, логістиці, екології, медицині, ІТ та інших галузях. Відповідно змінюються й освітні пріоритети: школа має не лише передавати готові знання, а формувати здатність учнів досліджувати, моделювати явища, перевіряти гіпотези, робити висновки на підставі даних і аргументовано обирати найкращий варіант дій. Саме тому інтеграція комп'ютерного експерименту в навчальний процес розглядається як одна з ключових тенденцій сучасної освіти, адже поєднує класичні способи пізнання з можливостями цифрових середовищ, симуляцій та обчислювального аналізу.

Особливого значення така інтеграція набуває під час вивчення моделювання та задач оптимізації. Моделювання дозволяє описувати реальні об'єкти й процеси в спрощеній, але керованій формі, виділяючи суттєві змінні, параметри та зв'язки між ними. Оптимізація, у свою чергу, дає методологію пошуку найкращого рішення за заданим критерієм за наявності обмежень, що відповідає логіці реальних ситуацій: ресурси завжди обмежені, а цілі — конкурують. У навчанні це означає перехід від пасивного засвоєння алгоритмів до діяльнісного підходу, коли учень не просто розв'язує задачу «за зразком», а аналізує умови, формалізує проблему, будує модель, проводить серію обчислювальних спроб і обґрунтовує вибір рішення. Комп'ютерний експеримент у цьому контексті є не «прикрасою» уроку, а методичною опорою, що забезпечує перевірку адекватності моделі, порівняння альтернатив, дослідження чутливості результатів до змін параметрів та формування навичок інтерпретації отриманих даних.

Актуальність дослідження зумовлена кількома взаємопов'язаними чинниками. По-перше, у межах STEM-освіти та компетентнісного підходу зростає частка завдань, де потрібні не лише обчислення «вручну», а й робота з цифровими інструментами, таблицями, графіками та програмними середовищами, що моделюють процеси та реалізують оптимізаційні алгоритми. По-друге, підготовка учнів до реального світу вимагає формування дослідницьких умінь: ставити проблему, висувати припущення, перевіряти їх на даних, робити узагальнення й оцінювати обмеження отриманого результату. Саме комп'ютерний експеримент дозволяє організувати навчальну діяльність так, щоб учень бачив наслідки зміни параметрів і розумів, чому певне рішення є оптимальним лише за конкретних умов. По-третє, у практиці шкільного викладання існує потреба в методично вивірених сценаріях, які б допомагали вчителю інтегрувати оптимізаційні задачі у курс моделювання без перевантаження теорією та без втрати наочності. По-четверте, доступність сучасних інструментів (електронні таблиці, освітні платформи, прості мови програмування, математичні пакети) створює реальні можливості для організації таких експериментів у межах закладів загальної середньої освіти, але потребує обґрунтованого відбору засобів і методик їхнього застосування.

Об'єкт дослідження — освітній процес у закладах загальної середньої освіти в контексті навчання математичного моделювання. Предмет дослідження — комп'ютерний експеримент у задачах оптимізації як методичний засіб підвищення ефективності навчання моделюванню.

Мета роботи полягає в обґрунтуванні доцільності використання комп'ютерного експерименту для розв'язання задач оптимізації під час вивчення моделювання в закладах загальної середньої освіти, а також у визначенні методичних умов, за яких така інтеграція забезпечує кращі навчальні результати.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. розкрити зміст і сутність математичного моделювання та окреслити підходи до класифікації моделей у навчальних цілях;

2. охарактеризувати основні методи оптимізації, показати їх зв'язок із моделлю та критерієм оптимальності, визначити дидактичні можливості використання в шкільному курсі;

3. обґрунтувати роль комп'ютерного експерименту як етапу навчального моделювання, що забезпечує перевірку, уточнення та інтерпретацію результатів;

4. здійснити аналіз програмних засобів і цифрових середовищ, придатних для використання у школі, та визначити критерії їхнього методичного відбору.

Методологічну основу дослідження становлять положення компетентнісного та діяльнісного підходів в освіті, а також ідеї STEM-навчання, що акцентують міжпредметність, практичну спрямованість та опору на експеримент. У процесі дослідження використано загальнонаукові й спеціальні методи: аналіз, синтез та порівняння науково-педагогічної і методичної літератури; узагальнення та систематизацію підходів до навчання моделюванню; метод моделювання для структурування етапів побудови та перевірки моделей; педагогічне спостереження за особливостями виконання учнями дослідницьких завдань; елементи експериментального підходу, пов'язані з організацією комп'ютерних прогонів, фіксацією результатів у табличній формі та їх подальшим аналізом.

Наукова новизна роботи полягає у впорядкуванні та дидактичній адаптації оптимізаційних задач для шкільного викладання з урахуванням можливостей комп'ютерного експерименту як навчального інструмента. У межах дослідження уточнено методичну логіку переходу від реальної проблеми до формалізації, плану експерименту, серії обчислювальних прогонів і подальшої інтерпретації, що дозволяє підсилити зв'язок між математичними поняттями та практичним змістом завдання.

Практичне значення отриманих результатів полягає в можливості їх безпосереднього використання у шкільній практиці. Матеріали дослідження можуть застосовуватися вчителями математики та інформатики для підготовки

уроків і навчальних проєктів із моделювання та оптимізації, у факультативній і гуртковій роботі, а також під час організації міжпредметних STEM-занять. Запропонований підхід може слугувати основою для добору програмних засобів, створення завдань із дослідницьким компонентом та підготовки учнів до роботи з даними, графіками й табличними моделями.

Обсяг і структура роботи зумовлені метою та завданнями дослідження. Робота складається зі вступу, двох розділів, висновків, списку використаних джерел і додатків, що містять ілюстративні матеріали та рисунки.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ КОМП'ЮТЕРНОГО ЕКСПЕРИМЕНТУ В ЗАДАЧАХ ОПТИМІЗАЦІЇ

1.1. Поняття та класифікація математичного моделювання

Математичне моделювання в сучасних задачах оптимізації розглядають як метод пізнання й проєктування, за якого реальний об'єкт, процес або система замінюється формалізованим описом у вигляді математичної моделі, а основні висновки отримують не шляхом прямого втручання в реальність, а шляхом аналізу та обчислювального експерименту над цим описом. У прикладному сенсі математичне моделювання є «мовою», що дозволяє перевести змістовну постановку задачі (цілі, ресурси, обмеження, критерії якості) у систему змінних, параметрів і співвідношень, придатних для розрахунків, порівняння альтернатив і обґрунтування управлінських рішень. Для задач оптимізації це особливо важливо, тому що оптимізація передбачає не просто відтворення поведінки системи, а цілеспрямований пошук таких значень керованих змінних, які забезпечують найкраще значення заданого критерію за наявних обмежень.

Поняття «модель» у науково-інженерному контексті доцільно визначати як умовний образ об'єкта, що відтворює істотні для дослідження властивості й зв'язки, відкидаючи другорядні деталі. Відповідно математична модель — це модель, виражена мовою математичних символів: змінних, функцій, рівнянь, нерівностей, операторів, імовірнісних розподілів, логічних умов. Її ключова риса полягає в тому, що вона одночасно має інтерпретацію в предметній області (економіці, техніці, логістиці, інформатиці) й допускає строгі операції аналізу: доведення властивостей, чисельні розрахунки, оцінювання чутливості, пошук оптимуму. Коли така модель реалізується у вигляді алгоритму та програмного коду, а дослідник варіює параметри й умови для отримання результатів, говорять про комп'ютерний (обчислювальний) експеримент. У задачах оптимізації комп'ютерний експеримент часто є головним інструментом перевірки гіпотез щодо структури моделі, коректності обмежень і стабільності оптимального рішення в умовах невизначеності.

Важливо підкреслити, що математичне моделювання не зводиться до «написання формул». Воно включає послідовні інтелектуальні дії: вибір меж системи та рівня абстракції, визначення змінних і параметрів, встановлення причинно-наслідкових або кореляційних зв'язків, формалізацію обмежень ресурсів і технологічних правил, вибір критерію оптимальності та узгодження одиниць вимірювання. У реальних задачах оптимізації помилка найчастіше виникає не на етапі обчислень, а на етапі концептуалізації: коли, наприклад, вартісний показник підміняє фізичні обмеження, або коли критерій «мінімізувати час» конфліктує з умовою «забезпечити якість» і це не відображено через багатокритеріальність чи штрафні функції. Тому у теоретичному описі моделювання акцент роблять на адекватності: модель має бути достатньо точною для поставленої мети, але не надмірно складною, щоб залишатися керованою для аналізу й оптимізації.

Класифікація математичних моделей потрібна не як формальна «типологія заради типології», а як інструмент вибору методів розв'язування, оцінювання складності та інтерпретації результатів. У контексті оптимізації класифікаційні ознаки безпосередньо впливають на те, чи можна застосувати точні методи (симплекс, внутрішньо-точкові методи, методи опуклої оптимізації), чи доведеться використовувати евристики та метаетристики (генетичні алгоритми, рій частинок), чи потрібні стохастичні підходи (Монте-Карло, стохастичне програмування), чи виправдане імітаційне моделювання із подальшою оптимізацією за результатами симуляції. На практиці одну й ту саму систему можна моделювати по-різному: транспортний процес можна описати лінійною моделлю потоків, дискретно-подієвою симуляцією черг або агентною моделлю поведінки учасників, і кожен варіант дає різні можливості оптимізації.

Однією з базових підстав класифікації є ступінь детермінованості. Детерміновані моделі припускають, що всі параметри та зв'язки задані однозначно, а однакові початкові умови завжди дають однаковий результат. В оптимізації це відповідає класичним постановкам лінійного, нелінійного, цілочисельного або динамічного програмування, де коефіцієнти відомі та

фіксовані. Стохастичні моделі, навпаки, вводять випадковість у параметри, зовнішні впливи або самі стани системи; тут оптимізація часто орієнтується на мінімізацію математичного сподівання витрат, ризику (наприклад, $VaR/CVaR$), або на забезпечення ймовірнісних обмежень. Різниця між цими класами визначає не лише математику, а й зміст управлінських висновків: детермінований оптимум може виявитися крихким, якщо реальні коливання попиту або часу обслуговування не враховані.

Друга фундаментальна ознака — часовий характер: статичні та динамічні моделі. Статична модель описує систему «в один момент» або в агрегованому вигляді без явного часу, що зручно для задач розподілу ресурсів, вибору портфеля, планування обсягів виробництва на період. Динамічна модель фіксує еволюцію станів у часі й використовує диференціальні рівняння, різницеві рівняння, марковські процеси, моделі керування. Для оптимізації це означає інший клас задач: з'являються траєкторії керування, обмеження на переходи, запізнення, накопичення запасів, а оптимальність визначається на горизонті планування. У прикладних системах динамічність часто є джерелом складності: навіть лінійні залежності можуть породжувати великі розмірності та потребу в спеціальних алгоритмах.

Третя важлива ознака — дискретність опису. Неперервні моделі оперують змінними, що можуть набувати будь-яких значень у діапазоні, і є природними для фізичних процесів, енергетики, задач керування, де витрати або потоки змінюються плавно. Дискретні моделі містять цілочисельні змінні, логічні умови, комбінаторні структури; у задачах оптимізації це типово для маршрутизації, розкладів, планування змін, вибору конфігурацій, розміщення об'єктів. Саме дискретність часто робить задачу NP-складною, тому в комп'ютерному експерименті значну роль відіграють евристики, розсічення, гібридні алгоритми та порівняння якості рішень на тестових наборах даних.

Четверта ознака — лінійність або нелінійність. Лінійні моделі мають структуру, де критерій і обмеження є лінійними функціями змінних; вони добре інтерпретуються, ефективно розв'язуються й масштабуються, тому широко

застосовуються в економіці, логістиці, мережевому плануванні. Нелінійні моделі відображають більш реалістичні ефекти насичення, взаємодії, залежності від квадратів або добутоків змінних, але потребують складніших методів і можуть мати локальні оптимуми. Для комп'ютерного експерименту це означає, що перевірка стійкості оптимального рішення, аналіз чутливості та обґрунтування глобальності оптимуму стають окремими дослідницькими задачами.

Окрема, практично значуща класифікація — за способом отримання моделі: механістичні (побудовані «від принципів», із законів і правил предметної області) та дано-орієнтовані (емпіричні, ідентифіковані зі спостережень). Механістична модель зручна для пояснення причинності й для екстраполяції за межі даних, але може бути складною та вимагати параметризації. Дано-орієнтована модель (регресійна, машинного навчання, сурогатна модель) часто дає високу точність прогнозу в межах навчального діапазону і може прискорити оптимізацію як наближення «дорогої» симуляції. У задачах оптимізації дедалі частіше застосовують гібридні підходи, коли фізично обґрунтована структура доповнюється статистичною оцінкою параметрів, а пошук оптимуму здійснюється з урахуванням невизначеності прогнозів.

Таблиця 1.1 Класифікаційна ознака та типи моделі

Класифікаційна ознака	Тип моделі	Коротка характеристика	Типові приклади в задачах оптимізації
Визначеність параметрів	Детермінована / стохастична	Фіксовані коефіцієнти або випадкові впливи/параметри	ЛП/НЛП, цілочисельна оптимізація / стохастичне програмування, оптимізація з ризиком
Часовий аспект	Статична / динамічна	Без явного часу або з еволюцією станів	Розподіл ресурсів за період / оптимальне

			керування, моделі запасів, планування на горизонті
Природа змінних	Неперервна / дискретна / змішана	Реальні або цілочисельні (логічні) змінні	Опукла оптимізація / маршрутизація, розклади / MILP, MINLP
Форма залежностей	Лінійна / нелінійна	Лінійні або нелінійні критерії та обмеження	ЛП, мережеві потоки / НЛП, оптимізація з насиченням, взаємодіями
Рівень агрегування	Макро / мікро	Узагальнений або детальний опис елементів	Агреговані балансові моделі / агентні, дискретно-подієві моделі з деталізацією
Спосіб побудови	Механістична / дано-орієнтована / гібридна	Від законів предметної області або з даних	Оптимізація техпроцесів / оптимізація за прогнозами ML / сурогатні моделі + фізичні обмеження
Метод дослідження	Аналітична / імітаційна	Розв'язування через формальний аналіз або через симуляцію	Оптимізація з гарантіями / оптимізація «симуляція + пошук» (Monte Carlo, DES)

Отже, класифікація математичних моделей у задачах оптимізації є практичним каркасом для вибору методології комп'ютерного експерименту. Вона дозволяє ще на етапі постановки визначити, які властивості рішення можна обґрунтувати строго, які доведеться підтверджувати серіями обчислювальних

експериментів, як інтерпретувати відхилення між моделлю та реальністю, а також які ризики пов'язані з невизначеністю параметрів і структурними спрощеннями. У наступних підрозділах логічним продовженням є розгляд комп'ютерного експерименту як процедури перевірки адекватності моделі, аналізу чутливості та пошуку оптимальних рішень у різних класах постановок, коли вибір типу моделі прямо визначає вибір алгоритмів, критерії якості та спосіб валідації отриманих результатів.

1.2. Основи оптимізації та їх застосування в освітньому процесі

Оптимізація в прикладній математиці та інформатиці зазвичай розуміється як формалізований пошук «найкращого» рішення серед множини допустимих рішень, коли «найкраще» визначається обраним критерієм якості. У класичній інтерпретації, яку подають С. Бойд і Л. Ванденберге, задача оптимізації задається змінною рішення (x), цільовою функцією (objective) та системою обмежень; розв'язком є допустимий вектор (x^*), що дає найменше значення цільової функції серед усіх допустимих рішень. У підручниках з чисельної оптимізації (зокрема у Дж. Носедала та С. Райта) цей самий зміст акцентується з прикладного боку: оптимізаційні задачі виникають тоді, коли потрібно мінімізувати/максимізувати показник якості за умов наявних ресурсних, технологічних або логічних обмежень, а форма функцій та обмежень визначає клас задачі й методи її розв'язування.

Як правило, будь-яку оптимізаційну постановку можна описати через три ключові компоненти. По-перше, це керовані змінні (decision variables) — те, чим ми можемо «керувати» у системі: розклад занять, розподіл аудиторій, вибір набору тестових завдань, призначення викладачів, порядок проходження навчальних модулів. По-друге, це цільова функція ($f(x)$), яка перетворює рішення на число і тим самим дозволяє порівнювати альтернативи: мінімізувати «вікна» в розкладі, зменшити конфлікти іспитів, вирівняти навантаження, підвищити точність вимірювання здібностей у тесті, максимізувати «покриття» навчальних

результатів тощо. По-третє, це обмеження, які фіксують реальні правила й ресурси: місткість аудиторій, доступність викладача, сумісність груп, тривалість занять, вимоги акредитації, умови валідності оцінювання. Саме така тріада — змінні, мета, обмеження — є «каркасом» оптимізації у трактуванні Бойда та Ванденберге, де також вводиться поняття допустимої множини (feasible set) і оптимального розв'язку як найкращого серед допустимих.

Далі оптимізацію класифікують за властивостями цільової функції та обмежень, бо від цього залежить і складність, і гарантії якості розв'язку. Лінійні задачі (лінійне програмування) мають лінійну ціль і лінійні обмеження; вони добре масштабуються і часто є «першим вибором» для управлінських задач в освіті. Якщо в моделі з'являються цілочисельні рішення (наприклад, «заняття або поставлено в слот, або ні», «питання або входить у тест, або ні»), задача стає дискретною/цілочисельною, а такі постановки описують як оптимальний розподіл обмежених ресурсів за наявності вимоги дискретності. Окремо виділяють опуклу оптимізацію, де опуклість забезпечує принципову зручність: локальний оптимум є глобальним, і це радикально підвищує надійність і швидкість знаходження розв'язку. У реальних освітніх задачах часто маємо змішані моделі (частина змінних неперервні, частина дискретні), багатокритеріальні постановки (одночасно «якість + справедливість + ресурсна ефективність») та стохастичні варіанти, коли попит/відвідуваність/часи виконання завдань є випадковими.

Управлінський сенс оптимізації в освіті найчастіше реалізується в логіці операційних досліджень (operations research, OR). IFORS визначає OR як процес прийняття кращих рішень через аналіз даних, математичне моделювання та оптимізацію; INFORMS у публічній комунікації подає близький зміст як застосування аналітичних методів для кращих рішень. Саме тому оптимізація в освітньому процесі — це не лише «тема курсу», а й практичний інструмент для адміністрації, методичних служб, центрів якості освіти та розробників цифрових освітніх платформ.

Найвідоміша прикладна зона — складання розкладів (school/university timetabling). У літературі це стабільно трактується як складна задача комбінаторної оптимізації: потрібно узгодити часові слоти, аудиторії, викладачів і групи так, щоб виконати обов'язкові обмеження (конфлікти, доступність, місткість) і водночас мінімізувати «м'які» штрафи (вікна, небажані пари, нерівномірність навантаження). Оглядові роботи підкреслюють різноманітність підходів — від математичного програмування до локального пошуку та метаевристик — і пояснюють, чому універсального «одного» методу немає: постановки відрізняються правилами і пріоритетами. Як окремий приклад, дослідження зі шкільного таймтейблінгу описують використання математичної моделі та процедур локального/ітеративного локального пошуку для побудови й покращення розкладів з урахуванням преференцій викладачів.

Друга важлива управлінська задача — розподіл викладацького навантаження. Вона добре лягає на лінійне програмування, бо є природні обмеження (кількість годин, компетентність/можливість викладання дисципліни, верхні та нижні межі навантаження), а ціль часто формулюють як мінімізацію «відхилення від справедливості» або максимізацію відповідності побажанням і кваліфікації. Приклад такого підходу описано у кейсі математичного департаменту, де LP-модель використана для оптимального розподілу контактних годин із урахуванням обмежень та преференцій.

Третя зона — оптимізація оцінювання, особливо в масштабних тестуваннях і комп'ютеризованих іспитах. Тут існує напрям automated test assembly (ATA), де тест «збирають» із банку завдань як задачу 0–1 лінійного програмування або змішано-цілочисельної оптимізації: кожне завдання або включається, або ні; обмеження задають змістові специфікації, час, рівні складності, баланс тем, а ціль може максимізувати точність вимірювання здібності. В оглядах і прикладних роботах прямо підкреслюється використання 0–1 LP/MILP як стандартного інструменту для ATA, часто разом із Item Response Theory як моделлю зв'язку між відповідями та латентною здібністю. Це важливо

для освітнього процесу, бо дозволяє підвищити валідність і порівнюваність форм тесту та зменшити ручну працю експертів без втрати контролю над змістом.

Четверта зона — персоналізація навчання в e-learning і системах рекомендацій навчального контенту. Тут оптимізація з'являється як задача вибору траєкторії навчальних кроків (learning path) за кількома критеріями: швидкість просування, досягнення результатів, мотивація/залученість, складність матеріалу, обмеження передумов (prerequisites). У сучасних публікаціях зустрічаються як евристичні, так і багатокритеріальні підходи (зокрема з використанням еволюційних алгоритмів), де мета — збалансувати ефективність і «людяні» показники навчання. У практиці закладу освіти це може бути реалізовано в LMS як «рекомендований маршрут» для різних профілів учнів на основі діагностичного тесту та історії взаємодії з контентом, але методично важливо, щоб критерії оптимальності були прозорі й педагогічно виправдані, інакше оптимізація перетворюється на непрозорий «скоринг», який важко пояснити учням та викладачам.

П'ята зона — стратегічні рішення та оцінювання ефективності закладів освіти. У цьому блоці OR застосовується ширше, ніж «розклади»: розглядаються питання фінансування, ресурсного планування, місткості, мережі закладів, оптимізації адміністративних процесів і вимірювання продуктивності/ефективності. Саме таку широку картину описує огляд «Operational Research in education», де наголошується, що освітній сектор породжує різні задачі для урядовців, менеджерів і споживачів освітніх послуг, а OR-техніки використовуються для підтримки рішень у різних підсистемах.

Таблиця 1.2 – Відповідність між типовими освітніми задачами та інструментарієм оптимізації, зручно узагальнити це в компактній таблиці.

Компонент освітнього процесу	Типова постановка	Природний клас моделі	Практичний результат

Розклад занять/іспитів	Мінімізувати конфлікти та «вікна» за жорстких обмежень	Дискретна/комбінаторна оптимізація; MILP; локальний пошук	Реалістичний розклад із керованими компромісами
Розподіл навантаження викладачів	Збалансувати години, врахувати компетентності й преференції	Лінійне програмування (LP)	Справедливіший розподіл годин, менше ручних конфліктів
Конструювання тестів (АТА)	Зібрати тест з банку завдань із вимогами до змісту і точності	0–1 LP / MILP + психометричні обмеження (IRT)	Валідніші форми, швидше складання, контроль специфікацій
Персоналізовані й маршрут навчання	Знайти траєкторію, що балансує ефективність, інтерес і складність	Багатокритеріальна оптимізація; евристики/еволюційні методи	Адаптивні рекомендації контенту з явними критеріями
Управлінські рішення в освіті	Планування ресурсів, політика, ефективність	OR/аналітика, оптимізація + моделювання	Краще обґрунтовані рішення для закладу/мережі

Окремо варто розрізняти «застосування оптимізації для управління освітою» і «викладання оптимізації в освітньому процесі». Для викладання важливе те, що оптимізація поєднує формалізацію реальної проблеми і комп'ютерний експеримент; це дає сильний навчальний ефект, коли учні бачать не лише формули, а й те, як змінюється рішення при зміні обмежень і пріоритетів. У педагогічній літературі з OR підкреслюється значення навчання постановці задач (formulation), інтерпретації результатів та написанню аналітичних звітів, а не лише «виведенню алгоритмів». Покажемо і приклад навчального модуля для

старшої школи, розробленого на базі OR-підходів, де мета — підвищити інтерес і мотивацію через реалістичні задачі прийняття рішень.

1.3. Роль комп'ютерного експерименту в навчанні моделюванню

У сучасній науці й інженерній практиці моделювання все частіше вивчають не як «теорію про моделі», а як повноцінну діяльність із власним циклом: від постановки задачі та формалізації припущень до побудови моделі, вибору чисельних методів, реалізації алгоритмів і перевірки отриманих результатів. Саме тут комп'ютерний (обчислювальний) експеримент стає ключовим дидактичним інструментом, бо дозволяє перетворити моделювання на керований навчальний процес із чіткою логікою «гіпотеза → перевірка → інтерпретація → уточнення моделі».

У науково-методичній традиції обчислювальний експеримент визначають як експеримент над математичною моделлю на комп'ютері, під час якого за одними параметрами моделі обчислюють інші й на цій підставі роблять висновки про властивості явища або процесу, що описуються моделлю. Це визначення важливе для педагогіки моделювання, тому що зміщує акцент із «виконання обчислень» на «логіку експериментування»: учень не просто рахує, а керує параметрами, спостерігає наслідки, фіксує закономірності та оцінює межі застосовності моделі. У тій самій роботі підкреслюється, що обчислювальний експеримент особливо цінний, коли натурний експеримент складний або неможливий, а також що він потребує осмислення адекватності моделі й інтерпретації даних, а не лише технічного запуску програми.

У більш широкому методологічному контексті обчислення і симуляції часто описують як «третю опору» наукового пізнання поряд із теорією та експериментом, бо комп'ютерні методи дають можливість будувати й перевіряти моделі складних систем, де аналітичні розв'язки недосяжні, а натурні випробування надто дорогі чи ризиковані. У популярно-науковій дискусії це

формулюють ще простіше: обчислення «закриває розрив» між теорією та експериментом, поєднуючи формальні припущення з перевіркою наслідків у керованому середовищі симуляції. Для навчання моделюванню це означає, що учню можна безпечно й швидко організувати багаторазові «прогони» моделі, змінюючи умови, порівнюючи сценарії та вчачись міркувати про причинно-наслідкові зв'язки в межах прийнятих припущень.

Дидактична цінність комп'ютерного експерименту полягає в тому, що він робить видимими ті компоненти моделювання, які в «чисто теоретичному» викладі часто залишаються прихованими: чутливість результату до параметрів, роль початкових умов, накопичення чисельних похибок, залежність від кроку дискретизації, межі стабільності алгоритму, різницю між верифікацією (чи правильно реалізовано обчислювальний метод) і валідацією (чи адекватно модель описує реальність у межах цілей дослідження). Коли учень бачить, що незначна зміна параметра може змінити поведінку моделі або зруйнувати збіжність, він починає сприймати моделювання як відповідальну процедуру побудови знання, а не як «калькулятор із формулами».

В університетській методиці навчання математики та моделювання комп'ютерний експеримент прямо пов'язують із мотивацією й наочністю: проведений у класі експеримент підвищує залученість і посилює візуалізацію, що критично при вивченні абстрактних понять; у самостійній роботі він працює на закріплення знань, поглиблення уявлень про об'єкт і опанування сучасних інструментів. Принципово важливим є й те, що експериментальна робота з гіпотезами «відкриває шлях» до справжньої дослідницької діяльності: учень отримує повний цикл — від формулювання проблеми до опису спостережень і їх аналітичної перевірки. Це прямо відповідає компетентнісній логіці сучасної освіти, де результатом є не відтворення визначень, а здатність самостійно будувати й обґрунтовувати модельні рішення.

Практика показує, що комп'ютерний експеримент найкраще працює в навчанні моделюванню тоді, коли його не підміняють «демонстрацією викладача», а проєктують як педагогічну ситуацію з керованим відкриттям. У

статті про комп'ютерний експеримент у викладанні математики наголошується, що під час детального планування доцільно виділяти мотивацію, власне експеримент, спостереження результатів, очікуване відкриття й гіпотези, а також подальше доведення гіпотез і перехід до неекспериментальної (формальної) математики. Для курсу моделювання це означає, що симуляція не замінює строгість, а створює місток до неї: учень спочатку «бачить» закономірність на даних і графіках, а потім вчиться її пояснювати мовою моделі та доведень, де це можливо.

Окремо варто підкреслити роль комп'ютерного експерименту саме у формуванні культури наукового мислення. Коли учень багаторазово запускає модель, він природно стикається з питанням інтерпретації: що саме ми вимірюємо в моделі, які змінні є спостережуваними аналогами реальних величин, які параметри є «керованими», а які — ідентифікованими з даних. Із цього виростає розуміння обмежень: обчислювальний експеримент дає детальну інформацію, але тільки в межах конкретної математичної моделі, тому він не «скасовує» натурні дані, а має поєднуватися з ними через процедури порівняння та уточнення. Саме цей момент критичний для навчання: учень має навчитися відрізняти переконливу картинку від обґрунтованого висновку.

Нижче подано узагальнення того, як комп'ютерний експеримент «вбудовується» в навчальний цикл моделювання і які компетентності формує.

Етап навчання моделюванню	Як проявляється комп'ютерний експеримент	Який навчальний результат формується
Постановка задачі та гіпотези	Учень формулює, що саме перевірятиме в моделі, і які параметри варіюватиме	Уміння операціоналізувати проблему й висувати перевірювані гіпотези
Побудова/вибір математичної моделі	Визначаються змінні, параметри, припущення, межі застосовності	Уміння обґрунтовувати припущення і бачити, як вони впливають на результат

Алгоритмізація та реалізація	Вибір чисельного методу, кодування, відтворення запуски	Уміння перетворювати модель на «обчислюваний об'єкт» і контролювати обчислювальну процедуру
Спостереження та візуалізація	Графіки, таблиці, анімації траєкторій, порівняння сценаріїв	Розвиток інтуїції щодо поведінки системи й уміння читати модельні дані
Перевірка та інтерпретація	Тести на граничні випадки, аналіз чутливості, зіставлення з очікуваннями/даними	Критичне мислення щодо результату, розрізнення «працює код» і «адекватна модель»
Узагальнення і перехід до теорії	Формулювання висновків, пояснення закономірностей, де можливо — доведення	Уміння переходити від емпіричного спостереження до строгого обґрунтування

Як підсумок, комп'ютерний експеримент у навчанні моделюванню виконує не допоміжну, а структуроутворюючу функцію. Він організовує навчання навколо дослідницьких дій учня, робить абстрактні моделі «керованими» та перевірюваними, формує відповідальне ставлення до припущень і похибок, а також створює міст між інтуїцією, даними й теоретичним поясненням. Саме тому в курсах, де моделювання пов'язане з оптимізацією (вибір найкращого рішення за критерієм і обмеженнями), комп'ютерний експеримент стає основним способом навчити учня не лише знаходити відповідь, а й доводити її коректність у межах прийнятої моделі та чесно описувати межі застосовності отриманих результатів.

1.4. Аналіз програмних засобів для проведення комп'ютерних експериментів.

Найпоширенішою «базовою» платформою є екосистема Python, де NumPy забезпечує масиви та високопродуктивні операції над ними [1], а SciPy додає

бібліотеку алгоритмів для оптимізації, інтегрування, розв'язування диференціальних рівнянь та інших класів задач [2]. Для задач оптимізації особливо важливим є модуль `scipy.optimize`, який прямо орієнтований на мінімізацію/максимізацію функцій (включно з обмеженнями) та споріднені підзадачі (root finding, least squares тощо) [3]. Перевага такого підходу для комп'ютерних експериментів полягає в тому, що учень або дослідник може швидко «замкнути цикл» від формули до серії прогонів і статистичного узагальнення, не розриваючи процес між різними інструментами.

Альтернативні (і часто комплементарні) платформи — R та Julia. R є вільним середовищем статистичних обчислень і графіки [4] та історично сильний у відтворюваній аналітиці й статистичному моделюванні (що важливо, коли комп'ютерний експеримент завершується перевіркою гіпотез). Julia позиціонується як мова високопродуктивних чисельних обчислень із поєднанням виразності високорівневого коду та продуктивності [5], що робить її привабливою для експериментів, де критичними є швидкість і масштабування, але водночас потрібна «наукова» зручність роботи з формулами.

Окремий клас становлять комерційні середовища MATLAB/Simulink. Simulink описується як блоково-діаграмне середовище для моделювання та симуляції, яке підтримує багатодоменні моделі й дозволяє проводити симуляцію до перенесення на апаратну платформу [6]. Такі системи особливо корисні в інженерній освіті, коли важливо навчити учнів зв'язувати математичні рівняння з «інженерною структурою» системи (контури керування, динаміка, сигнали), а також бачити поведінку моделі у часі в наочній формі.

Якщо комп'ютерний експеримент пов'язаний не стільки з розв'язуванням рівнянь, скільки з відтворенням поведінки складної системи (черги, логістика, взаємодія агентів, політики керування), тоді доцільні спеціалізовані симулятори. AnyLogic позиціонується як інструмент для симуляційного моделювання, який підтримує поєднання агентного підходу, дискретно-подійного моделювання та системної динаміки [7]; у документації/описах підкреслюється саме «мультиметодність» і можливість комбінувати системну динаміку з агентними та

подійними компонентами [8]. Для комп'ютерного експерименту це критично, оскільки дозволяє проводити сценарний аналіз: змінювати правила поведінки агентів, параметри потоків, політики обслуговування й порівнювати наслідки через стандартні показники (час очікування, пропускну здатність, завантаженість ресурсів).

Simulink у цьому блоці також розглядають як симуляційну платформу, але з іншим фокусом: він найбільш природний для безперервної динаміки та систем керування (тобто коли «експеримент» — це прогін моделі сигналів/станів у часі) [6]

У задачах оптимізації комп'ютерний експеримент часто означає не один запуск, а серію запусків моделі з різними сценаріями даних, параметрами штрафів, обмеженнями та політиками. Тут ключовими стають розв'язувачі (solvers) і бібліотеки OR. OR-Tools визначає себе як набір бібліотек і API для задач обмежувальної оптимізації, лінійної оптимізації та задач на графах/потоках [9]. Для освітніх кейсів OR-Tools зручний тим, що дає «вхід» у реальні задачі розкладів, маршрутів, призначень і дозволяє швидко переходити від постановки до експериментування зі сценаріями без надмірного математичного «важіння».

У випадках, коли потрібні промислові характеристики швидкодії та підтримка складних MIP/QP постановок, застосовують комерційні солвери на кшталт Gurobi. У довіднику Gurobi Optimizer зазначено, що це документація до оптимізатора, яка покриває моделювальні структури, можливості й API-концепції для практичного розроблення застосунків [10]. Освітньо-методично це дає змогу показувати учням «професійний стандарт» оптимізаційного інструментарію, але питання ліцензії й доступності слід враховувати окремо.

На рівні «легкого входу» в оптимізацію для експериментів корисний SciPy, оскільки `scipy.optimize` покриває широкий спектр задач оптимізації й дозволяє швидко порівнювати методи та чутливість розв'язку до параметрів [3]

Якість комп'ютерного експерименту в науковому сенсі визначається не лише отриманими числами, а здатністю іншого дослідника (або самого автора через час) повторити ті самі прогони в близькому середовищі з тими самими

налаштуваннями. Для цього використовують: інтерактивні лабораторні середовища (Jupyter), менеджери пайплайнів (Snakemake), контейнери (Docker) та трекінг експериментів (MLflow).

Jupyter-підхід цінний тим, що поєднує код, пояснювальний текст і вихідні артефакти в одному документі; у роботах, присвячених відтворюваному workflow, підкреслюється роль Jupyter як «виконуваного запису» обчислень і основи для керування computational study [11]. Для серійних експериментів і дисципліни відтворюваності важливим є перехід від «одного ноутбука» до керованого пайплайна. Snakemake прямо позиціонується як система керування workflow для створення відтворюваних і масштабованих аналізів, де кроки описуються декларативно, а сам workflow може масштабуватися на сервер/кластер без зміни визначення [12]. У задачах комп'ютерного експерименту це означає можливість формально задати серії прогонів (за параметрами), автоматично відтворювати результати та зменшувати ризик «ручних помилок» при повторних запусках.

Docker забезпечує фіксацію середовища виконання: документація підкреслює, що контейнери надають ізольовані відтворювані середовища, що дають передбачувані результати [13], а Dockerfile дозволяє формалізувати побудову образу середовища інструкціями [14]. Для навчання моделюванню це методично важливо, оскільки мінімізує ситуації, коли учень «не може повторити експеримент» через різні версії бібліотек.

Нарешті, якщо експерименти стосуються ML/параметричних запусків, актуальний клас інструментів — experiment tracking. MLflow Tracking описується як набір API та UI для фіксації параметрів, метрик і артефактів експериментів та подальшого порівняння запусків [15]. У навчальному процесі це дозволяє дисципліновано зберігати «історію експериментування» і пов'язувати висновки з конкретними параметрами, а не з пам'яттю виконавця.

Таблиця 1.1 – Порівняльна характеристика програмних засобів для комп'ютерних експериментів (моделювання та оптимізація)

Клас засобів	Представники	Типові задачі	Сильні сторони	Потенційні обмеження
--------------	--------------	---------------	----------------	----------------------

		комп'ютерного експерименту	(науково-методичні)	
Чисельні обчислення та аналіз даних	Python (NumPy, SciPy) [1–3] ; R [4]; Julia [5]	Параметричні прогони, Monte Carlo, оцінювання чутливості, чисельні методи	Висока гнучкість, широкий спектр алгоритмів, придатність до автоматизації експериментів	Потребує дисципліни відтворюваності (середовище, версії, seed)
Інженерне моделювання/симуляція	Simulink [6]	Динамічні системи, моделі керування, симуляція у часі	Наочність блокових схем, зручність для інженерної освіти	Комерційне ліцензування, «закритість» частини стеку
Симуляційні платформи складних систем	AnyLogic [7–8]	Дискретно-подійні процеси, агентні моделі, системна динаміка, сценарний аналіз	Мультиметодність, природність для «системних» кейсів (черги, потоки, взаємодії)	Поріг входу в моделювальні концепції; ліцензійні умови залежать від версії
Оптимізаційні бібліотеки/солвери	OR-Tools [9] ; Gurobi [10] SciPy optimize [3]	ЛП/ЦП/MIP, розклади, маршрутизація, призначення, сценарні прогони	Швидке порівняння політик і сценаріїв; підтримка реальних класів задач OR	Комерційні солвери потребують ліцензій; складні моделі потребують коректної постановки
Відтворюваність і керування workflow	Jupyter [11] ; Snakemake [12] Docker [13–14]	Автоматизація серій прогонів, фіксація середовища,	Відтворюваність, керованість експериментів, прозорість для перевірки	Потребує організаційної культури: структури проекту,

	MLflow [15]	журналювання метрик і артефактів		правил логування, контролю версій
--	-------------	----------------------------------	--	-----------------------------------

Ефективність програмного засобу для комп'ютерного експерименту визначається тим, наскільки він підтримує відтворюваний цикл «постановка → реалізація → серії прогонів → фіксація параметрів/метрик → інтерпретація». У навчанні моделюванню найбільш універсальною є комбінація відкритого обчислювального середовища (Python/NumPy/SciPy або R/Julia) з інструментами відтворюваності (Jupyter, Docker) та керування серіями запусків (Snakemake).

Таблиця 1.2 – Рекомендовані стеки інструментів залежно від типу навчального/наукового комп'ютерного експерименту

Тип експерименту в моделюванні	Мінімальний стек (навчальна лабораторія)	Розширений стек (курсний/дипломний рівень)	Пояснення вибору
Чисельне моделювання та статистичні прогони	Python + NumPy/SciPy [1–3] або R [4]	Jupyter [11] + Docker [13–14] (+ Snakemake [12])	Забезпечує повний цикл: реалізація → серії прогонів → відтворюваність
Оптимізація (лінійна/цілочисельна, розклади, маршрути)	OR-Tools [9] або SciPy optimize [3]	OR-Tools + (за доступності) Gurobi [10] + журналювання/реплікація середовища (Docker) [13–14]	Дозволяє проводити сценарний аналіз і порівняння політик на однакових даних
Динамічні системи (контури керування, сигнали)	Simulink [6]	Simulink + експорт даних/скриптовий аналіз (Python/R) [1–4]	Сильна наочність і зв'язок із інженерною

			інтерпретацією
Системна/агентна/подійна симуляція	AnyLogic [7–8]	AnyLogic + формалізований експеримент-план (серії сценаріїв, фіксація параметрів) + зовнішній аналіз даних	Зручно, коли «модель» — це система правил і взаємодій, а не тільки рівняння
ML/параметричні експерименти (як частина моделювання)	Python-стек [1–3]	MLflow [15] + Docker [13–14] (+ workflow-менеджер (Snakemake) [12] (	Критичним стає облік метрик, параметрів і артефактів для коректного порівняння прогонів

Для експериментів у системах керування та інженерній динаміці методично сильним є Simulink завдяки блоковій наочності й природному відображенню структури системи [6]. Для симуляції складних соціально-економічних або логістичних систем, де ключовими є події та взаємодії агентів, доцільні спеціалізовані симуляційні платформи (AnyLogic) із мультиметодним моделюванням [7–8]. Для оптимізаційних експериментів у задачах розкладів/маршрутизації логічно комбінувати OR-Tools як доступний навчально-прикладний інструмент [9] із промисловими солверами за наявності ліцензій і потреби в максимальній продуктивності [10], доповнюючи це контейнеризацією середовища та журналюванням результатів [13–15].

ВИСНОВОК ДО РОЗДІЛУ 1

Проведений аналіз теоретичних аспектів комп’ютерного експерименту в задачах оптимізації дозволяє зробити кілька важливих висновків щодо сутності, методів та застосування цього підходу. По-перше, математичне моделювання

виступає фундаментом для формалізації складних систем, на основі яких здійснюється оптимізація. Моделі дають можливість абстрагуватися від вторинних деталей та сконцентрувати увагу на ключових характеристиках системи, що спрощує процес дослідження та забезпечує високу точність прогнозів. Аналіз класифікацій моделей показав, що ефективне використання детермінованих, стохастичних, дискретних та безперервних моделей залежить від специфіки системи та поставлених завдань. Такий підхід дозволяє науковцям та практикам обирати оптимальні методи моделювання для різних ситуацій, забезпечуючи баланс між складністю моделі та доступними обчислювальними ресурсами.

По-друге, оптимізація як математичний та прикладний інструмент займає центральне місце в дослідженні систем. Вона дозволяє знаходити найкращі рішення у складних багатопараметричних умовах, що має важливе значення не лише у технічних та економічних задачах, а й у освітньому процесі. Використання методів оптимізації у навчанні дає змогу ефективно організовувати освітні ресурси, планувати навчальний процес та оцінювати ефективність різних стратегій навчання. Це забезпечує підвищення якості підготовки учнів та розвиток їх аналітичних і критичних навичок.

По-третє, роль комп'ютерного експерименту у навчальному процесі є надзвичайно важливою. Комп'ютерні експерименти дають можливість учням на практиці відпрацьовувати навички побудови математичних моделей, перевіряти гіпотези, здійснювати чисельні експерименти та аналізувати отримані результати. Це сприяє формуванню в учнів цілісного розуміння процесів моделювання та оптимізації, розвитку критичного мислення та вміння приймати обґрунтовані рішення на основі отриманих даних. Використання комп'ютерного експерименту також забезпечує безпечне тестування різних сценаріїв і параметрів системи, що особливо цінно у випадках, коли реальні експерименти складні або затратні.

Крім того, проведено аналіз сучасних програмних засобів для проведення комп'ютерних експериментів. Було встановлено, що сучасне програмне

забезпечення дозволяє ефективно моделювати складні процеси, проводити оптимізацію параметрів та здійснювати візуалізацію результатів. Інтерактивність таких засобів, наявність інструментів для графічного та чисельного аналізу, а також можливість швидкого порівняння різних сценаріїв роблять їх незамінними у навчанні та наукових дослідженнях. Застосування програмних продуктів дозволяє учням не лише розвивати практичні навички, а й глибше розуміти теоретичні аспекти моделювання та оптимізації, що підвищує ефективність навчального процесу.

Таким чином, теоретичні аспекти комп'ютерного експерименту створюють надійну основу для подальшого практичного використання методів моделювання та оптимізації. Вони забезпечують можливість гнучкого та обґрунтованого вибору методів дослідження, сприяють підвищенню ефективності навчання та формуванню в учнів комплексного бачення процесів моделювання. Комп'ютерний експеримент стає не лише інструментом дослідження, а й засобом розвитку аналітичного та критичного мислення, уміння прогнозувати наслідки змін параметрів та приймати оптимальні рішення.

Отже, результати розділу підтверджують, що інтеграція комп'ютерного експерименту в навчальний процес та наукові дослідження значно підвищує якість підготовки фахівців, забезпечує ефективне використання ресурсів та створює можливості для системного та комплексного аналізу складних систем. Такий підхід є перспективним напрямом розвитку сучасної освіти та науки, оскільки поєднує теоретичну підготовку, практичне навчання та сучасні технології обчислювальної техніки.

РОЗДІЛ 2. КОМП'ЮТЕРНИЙ ЕКСПЕРИМЕНТ В ЗАДАЧАХ ОПТИМІЗАЦІЇ

2.1. Розробка та апробація навчальних завдань з оптимізації

Практична частина побудована як серія навчальних міні-досліджень, у яких оптимізаційна задача подається через життєвий сюжет, далі трансформується у математичну модель (змінні, обмеження, цільова функція), а потім перевіряється та “досліджується” засобами комп'ютерного експерименту (електронні таблиці, графіки, надбудова Solver). Такий підхід забезпечує зв'язок між математичною постановкою і обчислювальною реалізацією, а також дає підстави для апробації завдань у форматі лабораторних/практичних робіт із чіткими критеріями оцінювання результату та коректності моделі. Для реалізації оптимізації в електронних таблицях доцільно використовувати надбудову Solver, оскільки вона підтримує цільову функцію, систему обмежень, а також цілочислові змінні, що особливо важливо для прикладних задач із “кількістю рейсів/виробів”. [1]

Розглянемо серію задач, що охоплює 6–9 класи та демонструє поступове ускладнення від параметричного пошуку максимуму однієї змінної до лінійного програмування та прогнозування за дискретними моделями. Перша задача (6 клас) має максимально наочний зміст і не потребує складного апарату: для шкільної клумби/майданчика необхідно спроектувати прямокутну огорожу так, щоб за заданої довжини огорожі площа була найбільшою. В умові фіксується периметр прямокутника $P = 148\text{м}$, а учневі пропонується визначити такі довжину та ширину, за яких площа прямокутника максимальна, і пояснити, чому саме ці розміри є оптимальними. У межах комп'ютерного експерименту задача реалізується як табличний перебір значень однієї змінної (наприклад, ширини b) з автоматичним обчисленням другої змінної з обмеження периметра та побудовою графіка залежності площі S від b ; оптимум визначається як найбільше значення у таблиці/на графіку, а потім уточнюється зменшенням кроку дискретизації.

Друга задача (7 клас) має геометричний контекст і спрямована на формування навички переходу від рисунка до формули. Із круглої колоди радіуса r вирізають балку прямокутного перерізу найбільшої площі; потрібно знайти розміри перерізу балки. В умові задається радіус колоди, зокрема для демонстраційного прогону береться $r = 20\text{см}$, після чого учні встановлюють зв'язок між сторонами прямокутника a і b , вписаного в коло: діагональ прямокутника дорівнює діаметру колоди, тому виконується обмеження $a^2 + b^2 = (2r)^2$. Потрібно визначити такі a та b , що площа $S = ab$ є максимальною, і підтвердити результат комп'ютерним експериментом через параметризацію (табличний прогін за a із кроком дискретизації та обчисленням b з обмеження, побудова графіка $S(a)$) або через динамічні середовища на кшталт GeoGebra.

Третя задача (8 клас) переводить оптимізацію у площину прикладного планування з обов'язковою цілочисловістю. Потрібно організувати перевезення двох типів вантажу M та S з мінімальною загальною кількістю рейсів. Загальний обсяг перевезення становить $K_m = 15\text{т}$ вантажу типу M та $K_s = 15\text{т}$ вантажу типу S . Доступні три типи рейсів, причому в одному рейсі першого типу перевозиться 3 т M і 1 т S , у рейсі другого типу перевозиться 2 т M і 2 т S , у рейсі третього типу перевозиться 1 т M і 4 т S . Необхідно визначити, скільки рейсів кожного типу слід виконати, щоб виконати вимоги за обома видами вантажу та мінімізувати сумарну кількість рейсів, причому числа рейсів мають бути невід'ємними цілими. У комп'ютерному експерименті ця задача є природною для Solver: цільова функція — сума змінних (кількість рейсів), обмеження — рівності/нерівності на перевезені обсяги M і S , додаткова умова — цілочисловість змінних.

Четверта задача (9 клас) представляє класичну постановку «оптимальний план випуску продукції» та демонструє застосування лінійного програмування в управлінському контексті. Кондитерський цех виготовляє чотири види продукції: пиріжки (максимально 2000 шт. на день), бублики (максимально 1500 шт. на день), булочки (максимально 1200 шт. на день), тістечка (максимально 700 шт. на

день). Відомі співвідношення вартості: пиріжок/булочка = 2/1, пиріжок/бублики = 1/2, пиріжок/тістечко = 1/4, а також обмеження складу: загальна місткість складу дорівнює 1100 виробам незалежно від виду. Потрібно скласти оптимальний план випуску, що забезпечує найбільшу виручку за дотримання всіх обмежень. У моделі вводяться змінні як кількості виробів кожного виду, цільова функція задається через виручку (ціни можуть бути нормовані відносно, наприклад, ціни булочки), а обмеження включають верхні межі випуску та складську місткість; при потребі додається вимога цілочисловості, оскільки кількість виробів є дискретною.

П'ята задача (9 клас) відноситься до блоку прогнозування та показує, що оптимізаційно-експериментальна логіка може бути використана також для сценарного аналізу параметрів моделі. Розглядається екологічна система «два хижаки — одна жертва», де початкова чисельність популяції мишей становить 5000 особин; мишами живляться лисиці та сови. Початкова чисельність лисиць становить 20 особин, і за рік одна лисиця поїдає в середньому 40 мишей; початкова чисельність сов становить 15 особин, і за рік одна сова поїдає в середньому 30 мишей. Для кожної популяції задаються параметри річного приросту та смертності: для мишей ріст 0,5 і смертність 0,1; для лисиць ріст 0,13 і смертність 0,05; для сов ріст 0,1 і смертність 0,03. Потрібно розрахувати чисельність популяцій мишей, лисиць і сов через 5 і через 10 років, використовуючи дискретну (рекурентну) модель, а також провести сценарний експеримент, змінюючи окремі параметри й аналізуючи, як це впливає на стійкість системи та можливі критичні режими (наприклад, «вимирання» одного з видів при надмірній смертності або надто високому споживанні). В Excel ця задача реалізується як таблиця років із рекурентними формулами та діаграмою трендів, де результатом є не лише числа, а й пояснення характеру динаміки.

Таблиця 2.1 – Навчальні завдання з оптимізації (6–9 класи): сюжет, метод і цифровий інструмент

Клас	Тематика оптимізації	Математична ідея	Інструмент комп'ютерного експерименту	Очікуваний результат навчання
6	Максимізація площі за фіксованої довжини огорожі	Одна змінна; цільова функція площі; табличний перебір	Excel: таблиця значень + графік	Учень уміє знаходити максимум за таблицею/графіком і пояснювати залежність
7	Оптимальна балка з круглої колоди (вписаний прямокутник)	Геометричне обмеження $a^2 + b^2 = 4r^2$; максимум площі	Excel або GeoGebra: параметризація + пошук максимуму	Учень переводить геометричну умову в формулу та досліджує максимум
8	Мінімізація кількості рейсів (перевезення комплектів)	Лінійні обмеження + цілочислова мінімізація	Excel Solver (цілочисловий режим)	Учень формалізує задачу як ILP і пояснює отриманий план
9	Оптимальний план випуску продукції (виручка/ресурси)	Лінійне програмування (max)	Excel Solver (Simplex LP)	Учень будує LP-модель і інтерпретує план як управлінське рішення
9	Прогнозування в екології (“два хижаки – одна жертва”)	Рекурентні співвідношення; сценарний аналіз параметрів	Excel: рекурентні формули + діаграма трендів	Учень реалізує дискретну модель і робить висновки про динаміку системи

Сюжет завдання 1 (клас) ґрунтується на моделюванні обгородженої території, де огорожі вистачає лише на три сторони (четверта сторона — природна межа). У поданому навчальному матеріалі показано, що при довжині огорожі L оптимізація зводиться до вибору ширини b (керуючої змінної), після чого довжина a визначається як $a = L - 2b$, а площа $S = ab$ максимізується.

Таблиця 2.2 – Алгоритм комп’ютерного експерименту в Excel (6 клас): пошук максимуму площі $S = ab$

Крок	Дія в Excel	Зміст математичної дії
1	Увести параметр L (наприклад, у комірку B1)	Фіксація ресурсу (довжина огорожі)
2	Створити стовпчик значень b (наприклад, 0; 0,5; 1; ...; $L/2$)	Дискретизація керуючої змінної
3	У сусідньому стовпчику обчислити $a = L - 2b$	Реалізація обмеження $L = 2b + a$
4	Обчислити $S = a \cdot b$	Побудова цільової функції $S \rightarrow \max$
5	Знайти максимум S (функцією MAX або сортуванням) та побудувати графік $S(b)$	Виявлення аргументу максимуму та його інтерпретація

У шкільній апробації для 6 класу доцільно задати прості числа (наприклад, $L = 30$ м або інше значення, узгоджене з рівнем класу), щоб зручно виконувати табличний перебір. Важливо, що навіть без похідних учні отримують “дослідницький” результат: максимум площі досягається в конфігурації, коли довжина удвічі більша за ширину, тобто $a \approx 2b$. Ця закономірність прямо в частині дипломної роботи, як узагальнений висновок комп’ютерного експерименту.

У навчальному матеріалі оптимізація подається як задача про прямокутний переріз балки (задача 2, 7 клас) , вписаний у коло радіуса r (переріз колоди). Керуючою змінною пропонується брати одну зі сторін прямокутника, а інша визначається з геометричного обмеження: сторони прямокутника є катетами прямокутного трикутника, гіпотенуза якого — діаметр кола.

На цій основі формується модель максимізації площі $S = ab$ за обмежень $0 < a < 2r$ та залежності $b = \sqrt{4r^2 - a^2}$ (у матеріалі записано як $b = 4r^2 - a^2$, що інтерпретується як підкореневий вираз).

Таблиця 2.3 – Алгоритм комп’ютерного експерименту (7 клас): максимізація площі вписаного прямокутника

Крок	Дія	Результат для моделі
1	Задати r (наприклад, 20 см)	Фіксація параметра системи
2	Задати сітку значень a на $(0; 2r)$	Дискретизація керування
3	Обчислити $b = \sqrt{4r^2 - a^2}$	Реалізація геометричного обмеження
4	Обчислити $S = a \cdot b$ і знайти максимум	Оцінка оптимального перерізу
5	Інтерпретувати закономірність: оптимум наближається до квадрата	Узагальнення результатів експерименту

Для 7 класу (№2) методично виправдано розв’язувати задачу не аналітично, а як комп’ютерний експеримент: учні будують таблицю значень a , обчислюють b та площу S , після чого знаходять максимум. Приклад для $r = 20$ см демонструє близькі значення сторін (практично квадрат) і підкреслює, що звуження кроку дискретизації підвищує точність.

Модель природно набуває вигляду цілочислового лінійного програмування: змінні P_1, P_2, P_3 — кількість рейсів із відповідним типом завантаження; цільова функція $P = P_1 + P_2 + P_3 \rightarrow \min$; обмеження балансу для моніторів та системних блоків; додатково — цілочисловість змінних.

У матеріалі прямо зазначено, що задачу доцільно розв’язувати надбудовою “Пошук рішень” (Solver) і наведено контрольний результат: загалом 7 рейсів, причому 3 рейси за першим варіантом, 2 — за другим і 2 — за третім.

Таблиця 2.4 – Налаштування Solver для цілочислової оптимізації (8 клас)

Елемент Solver	Як задається	Чому це важливо
Цільова комірка	Комірка з формулою $P_1 + P_2 + P_3$	Мінімізує кількість рейсів
Змінні	Три комірки P_1, P_2, P_3	Керування планом перевезення
Обмеження	“Монітори = 15”, “Системні блоки = 15”	Виконується умова перевезення
Додаткове обмеження	P_1, P_2, P_3 — цілі	Забезпечує фізичний зміст (рейси не дробові)

Оскільки робота зі Solver вимагає попереднього підключення надбудови, у методичних вказівках до лабораторної роботи доцільно фіксувати стандартний шлях її ввімкнення в Excel через меню надбудов. [2]

Ця задача (4 ,8 клас) орієнтована на формування вміння будувати лінійно-оптимізаційні моделі управлінського типу. В умові задано 4 види продукції, ціни реалізації та ресурсні обмеження (борошно, яйця, цукор, бюджет).

Змінні x_1, x_2, x_3, x_4 інтерпретуються як обсяги випуску відповідних виробів (у кілограмах/партіях — залежно від узгодження одиниць у таблиці), а цільова функція — максимізація виручки $Z = 250x_1 + 340x_2 + 380x_3 + 290x_4$.

Обмеження записуються як лінійні нерівності за ресурсами та невід’ємність змінних.

Практична реалізація в Excel виконується через структуру “матриця коефіцієнтів → SUMPRODUCT → Solver”: у рядках задаються ресурси, у

стовпчиках — вироби, а сумарне споживання ресурсу обчислюється як скалярний добуток “норм витрат” і “плану випуску”. Далі Solver максимізує Z за заданих обмежень, використовуючи Simplex LP як стандартний метод для задач лінійного програмування. [1]

У навчальному матеріалі наведено контрольне оптимальне рішення: $x_1 = 100$, $x_2 = 50$, $x_3 = 20$, $x_4 = 40$, а максимальна виручка $Z = 59600$ грн.

Саме цей результат доцільно використовувати як критерій самоперевірки коректності постановки Solver (помилки найчастіше виникають у знаках нерівностей, посиланнях на комірки SUMPRODUCT або в одиницях виміру).

Таблиця 2.5 – Структура Excel-моделі для задачі планування випуску (9 клас)

Блок таблиці	Зміст	Приклад формули
Вхідні дані	Норми витрат ресурсів на вироби, доступні ресурси, ціни	Значення з умови
Змінні	Комірки $x_1 \dots x_4$	Початкові наближення (0 або реалістичні)
Цільова функція	Комірка виручки Z	$=\text{SUMPRODUCT}(\text{ціни}; x_1:x_4)$
Ресурсні обмеження	Використання кожного ресурсу	$=\text{SUMPRODUCT}(\text{рядок_норм}; x_1:x_4)$
Solver	Max Z при “використання $\leq$ запас”	Simplex LP [1]

Окремий блок 9 класу доцільно присвятити прогнозуванню, оскільки воно поєднує інформатику (табличні обчислення, формули, діаграми) та математику (рекурентні залежності як дискретний аналог диференціальних моделей). У поданій лабораторній роботі початкові чисельності такі: миші $M_0 = 5000$, лисиці $L_0 = 20$, сови $S_0 = 15$, а також задано відсотки приросту/смертності та коефіцієнти “поїдання” (лисиця — 40 мишей/рік, сова — 30 мишей/рік).

Математична модель подається системою рекурентних співвідношень для $i \geq 1$: $L_i = L_{i-1}(1 + P_l - S_l)$, $S_i = S_{i-1}(1 + P_s - S_s)$, $M_i = (M_{i-1} - L_i \cdot K_l - S_i \cdot K_s)(1 + P_m - S_m)$.

Саме цей запис зручний для Excel, тому що кожен новий рядок таблиці обчислюється через попередній, а результати легко візуалізуються діаграмою.

Таблиця 2.6 – Покрокова реалізація рекурентної моделі в Excel (9 клас)

Крок	Що робить учень	Як фіксується в Excel
1	Створює структуру таблиці “Рік – Миші – Лисиці – Сиви” для $i = 0..10$	Стовпчик років 0–10; три стовпчики популяцій
2	Вводить параметри $P_m, S_m, P_l, S_l, P_s, S_s, K_l, K_s$ окремим блоком	Наприклад, у комірках із підписами; значення з умови
3	Записує початкові значення M_0, L_0, S_0	Рядок для року 0
4	Обчислює L_1 і S_1 за формулами росту/смертності	$=L_0*(1+P_l-S_l)$; $=S_0*(1+P_s-S_s)$
5	Обчислює M_1 з урахуванням вилучення мишей хижаками та множника $(1+P_m-S_m)$	$=(M_0-L_1*K_l-S_1*K_s)*(1+P_m-S_m)$
6	Протягує формули до року 10 та буде лінійний графік трьох рядів	Діаграма типу “Лінійна” для M_i, L_i, S_i
7	Виконує сценарний аналіз, змінюючи один параметр і порівнюючи траєкторії	Наприклад, збільшення K_l або смертності S_m та повторне обчислення

Для методичного контролю коректності учнівської моделі доцільно використати наведену в матеріалі контрольну таблицю значень, де показано, що через 10 років чисельність мишей зменшується до 511, лисиць зростає до 35, а совів — до 25 (за параметрами задачі).

Таблиця 2.7 – Критерії оцінювання навчального комп’ютерного експерименту з оптимізації

Компонент результату	Ознака сформованості	Типові помилки, що виявляються під час апробації
Математична модель	Змінні, обмеження, цільова функція відповідають умові	Плутанина “max/min”, пропущене обмеження невід’ємності або цілочисловості
Таблична реалізація	Формули посилаються на правильні комірки, одиниці узгоджені	Неправильні посилання, змішані відсотки (0,13 vs 13)
Розв’язання (Solver/перебір)	Отримано оптимум, перевірено обмеження	Невірний метод, відсутні/неправильні обмеження
Інтерпретація	Рішення пояснено мовою сюжету (площа, рейси, випуск, популяції)	Підміна змісту “числом без сенсу”, відсутність висновку
Узагальнення	Сформульовано закономірність або висновок про чутливість	Ігнорування впливу параметрів; висновки без опори на дані

Розроблені навчальні завдання апробовано у форматі коротких практичних робіт з елементами комп’ютерного експерименту; детальні плани-конспекти цих робіт для 6–9 класів подано в додатках А–Г (див. Додаток А – 6 клас; Додаток Б – 7 клас; Додаток В – 8 клас; Додаток Г – 9 клас).

У задачах 8–9 класів критично важливо перевіряти постановку Solver (зокрема тип оптимізації, знаки нерівностей, наявність обмеження цілочисловості), що узгоджується зі стандартним підходом використання Solver для задач оптимізації в Excel. [1–2]

2.2 Використання комп'ютерного експерименту на уроках інформатики та математики на прикладі інтерактивного застосунку “School Optimization Experiment”

Комп'ютерний експеримент у шкільному курсі моделювання доцільно розглядати як керовану серію запусків моделі з варіюванням параметрів, під час якої учень не лише отримує числовий результат, а й проходить повний цикл “побудова моделі → обчислювальна перевірка → інтерпретація → уточнення припущень”. У сучасній дидактиці математичного моделювання підкреслюється циклічність процесу, де формалізація реальної ситуації переходить у дослідження, перевірку адекватності й тлумачення результатів у контексті задачі. Для математики й інформатики принципово важливо, що технології змінюють не тільки спосіб подання матеріалу, а й спектр математичних практик, які стають доступними учням: від візуалізації поведінки функції до експериментального дослідження області допустимих рішень та аналізу обмежень. Саме цю логіку реалізує розроблений інтерактивний застосунок “School Optimization Experiment”, який поєднує дві навчальні задачі оптимізації, табличну репрезентацію експериментальних прогонів, графічний супровід і окремий модуль інтерпретації результатів. Така організація роботи узгоджується з підходом дослідницького навчання, де “математичний/комп'ютерний експеримент” фактично виконує роль інструмента спостереження за моделлю через зміну ключових параметрів і аналіз отриманої поведінки, а також із позицією про педагогічну цінність комп'ютерного математичного експерименту як методики навчання на різних рівнях освіти.

Застосунок реалізовано як настільну програму мовою Python із графічним інтерфейсом на основі Tkinter та вбудованою віконною візуалізацією Matplotlib. Програмний дизайн навмисно зроблено “прозорим” для навчальних цілей: учень бачить керовані параметри моделі, отримує таблицю результатів як аналог електронної таблиці, спостерігає графік, а після цього відкриває окреме вікно інтерпретації, де пояснюється зміст оптимуму, похибка дискретного

експерименту або роль активних обмежень. Така послідовність дій важлива дидактично, оскільки зменшує ризик механічного “натискання кнопок” і переводить увагу на зміст моделі та причинно-наслідкові зв’язки між параметрами й результатом. У межах міжпредметної інтеграції інформатика забезпечує розуміння алгоритму перебору, подійної моделі GUI, структурування даних, експорт у CSV і відтворюваність експерименту, тоді як математика забезпечує формалізацію задачі (змінні, обмеження, цільова функція), інтерпретацію графіка та змістовний аналіз оптимального рішення. Підкреслимо, що подібна інтеграція відповідає сучасним орієнтирам розвитку математичної грамотності та обчислювального мислення як наскрізних компетентностей.

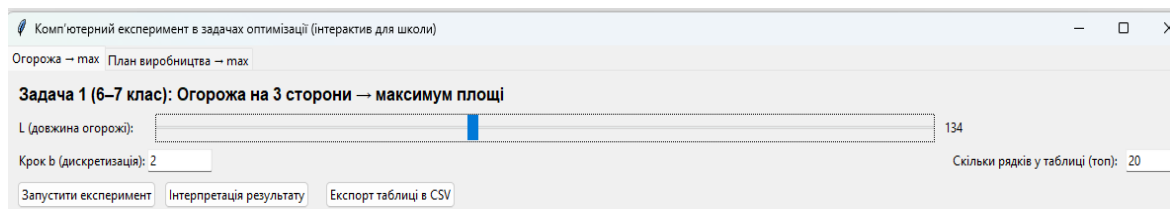


Рис. 2.1 – Головне вікно застосунку “School Optimization Experiment” із вкладками задач оптимізації (огорожа; план виробництва)

Перша вкладка застосунку присвячена задачі для 6–7 класів, яка має інтуїтивний сюжет і природно вводить поняття максимізації функції на відрізьку через комп’ютерний експеримент. Учні пропонується модель огорожі ділянки, де одна сторона прилягає до річки і не потребує огорожі, а на три інші сторони витрачається задана довжина матеріалу L . Модель задається обмеженням $L = a + 2b$, де a – довжина, b – ширина ділянки, а площа $S = a \cdot b$. Після підстановки $a = L - 2b$ маємо цільову функцію $S(b) = b(L - 2b)$, яку потрібно максимізувати. Застосунок реалізує комп’ютерний експеримент як дискретний прогін значень b із кроком, заданим учнем або вчителем, автоматично обчислює відповідні a та S , формує таблицю найкращих значень і будує графік $S(b)$, де точка максимуму позначається окремо. Дидактично суттєво, що учень отримує можливість змінювати не лише L , а й крок дискретизації, спостерігаючи ефект

збіжності: що менший крок, то ближче експериментальний оптимум до теоретичного, а похибка стає меншою. У такий спосіб комп'ютерний експеримент виконує функцію “контрольованої перевірки” та водночас вводить поняття похибки як властивості обчислювальної процедури, що формує коректне уявлення про обмеження чисельних методів у моделюванні.

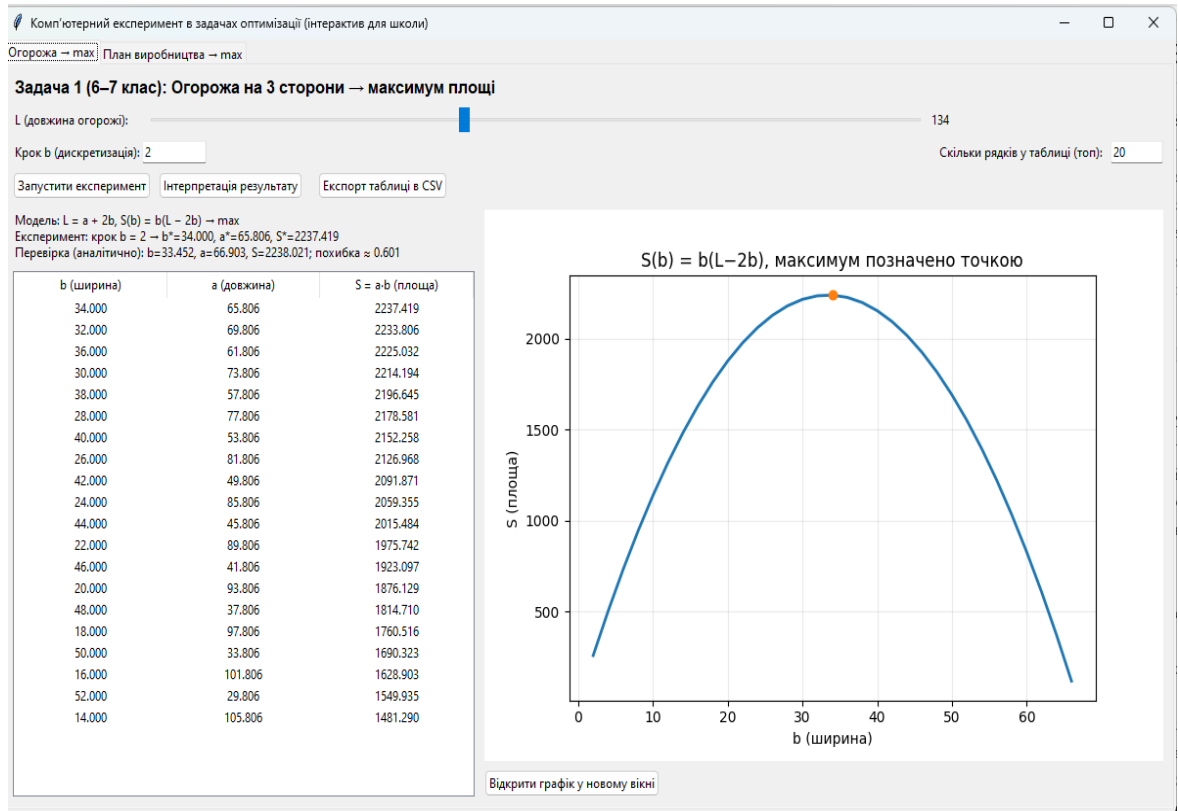


Рис. 2.2 – Вкладка “Огорожа → max”: таблиця експериментальних значень (b, a, S) та графік $S(b)$ із позначенням максимуму

Ключовим елементом цієї вкладки є вікно інтерпретації, яке відкривається окремою кнопкою та генерує зв'язний текст пояснення результату. У цьому тексті формулюється, що означає оптимум у термінах реальної ситуації, які змінні є керованими, чому максимум єдиний, як графічна форма параболи пояснює структуру оптимального розв'язку, а також як саме крок дискретизації впливає на точність знаходження найкращого значення. Такий модуль інтерпретації важливий методично, оскільки спрямовує учня на осмислення результатів і відповідає вимозі навчати не лише “обчислювати”, а й тлумачити математичні результати в контексті задачі , .

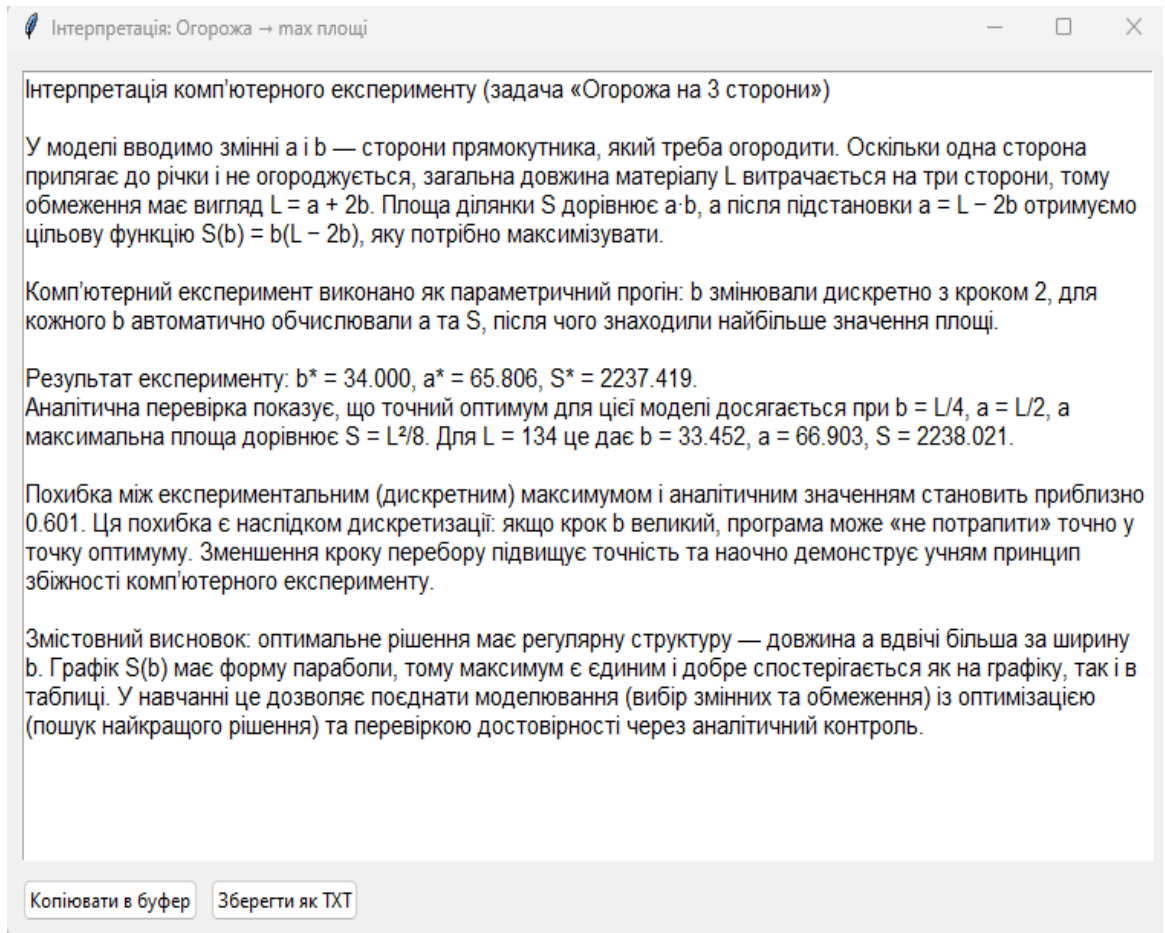


Рис. 2.3 – Вікно “Інтерпретація результатів” для задачі “Огорожа”:
пояснення оптимуму, впливу кроку та похибки дискретного експерименту

Друга вкладка застосунку орієнтована на 8–9 класи й ілюструє перехід від оптимізації однієї змінної до задачі з двома змінними та обмеженнями ресурсів, що наближає навчальний зміст до ідей лінійного програмування на рівні доступному школі. У задачі “план виробництва” вводяться змінні x_1 і x_2 як кількості двох типів виробів, цільова функція $Z = p_1x_1 + p_2x_2$ як сумарний результат (прибуток або виручка), а також два ресурсні обмеження виду $a_{11}x_1 + a_{12}x_2 \leq R_1$ і $a_{21}x_1 + a_{22}x_2 \leq R_2$. У комп'ютерному експерименті застосовано перебір допустимих пар (x_1, x_2) у цілочисловому режимі або на сітці заданого кроку, що дозволяє адаптувати складність до рівня класу. Результат подається в таблиці як рейтинг найкращих допустимих планів, а на графіку відображається область допустимих рішень і точка оптимуму. На відміну від суто “формульних” вправ, тут учень бачить, що оптимум не є “магічним числом”, а виникає через

конфлікт цілі та обмежень; зміна параметрів ресурсів або норм витрат закономірно змінює оптимальний план. Це створює підстави для змістовного обговорення: які обмеження стали лімітуючими, чому один ресурс може залишатися з запасом, а інший повністю вичерпується, і як це впливає на можливість подальшого зростання результату. У дослідженнях, присвячених комп'ютерному математичному експерименту, підкреслюється, що саме така візуалізація та робота з параметрами посилює розуміння структури математичного об'єкта й сприяє переходу від репродукції до дослідницьких дій, а інтеграція технологій у математичне навчання є принципово необхідною для забезпечення сучасних освітніх практик.

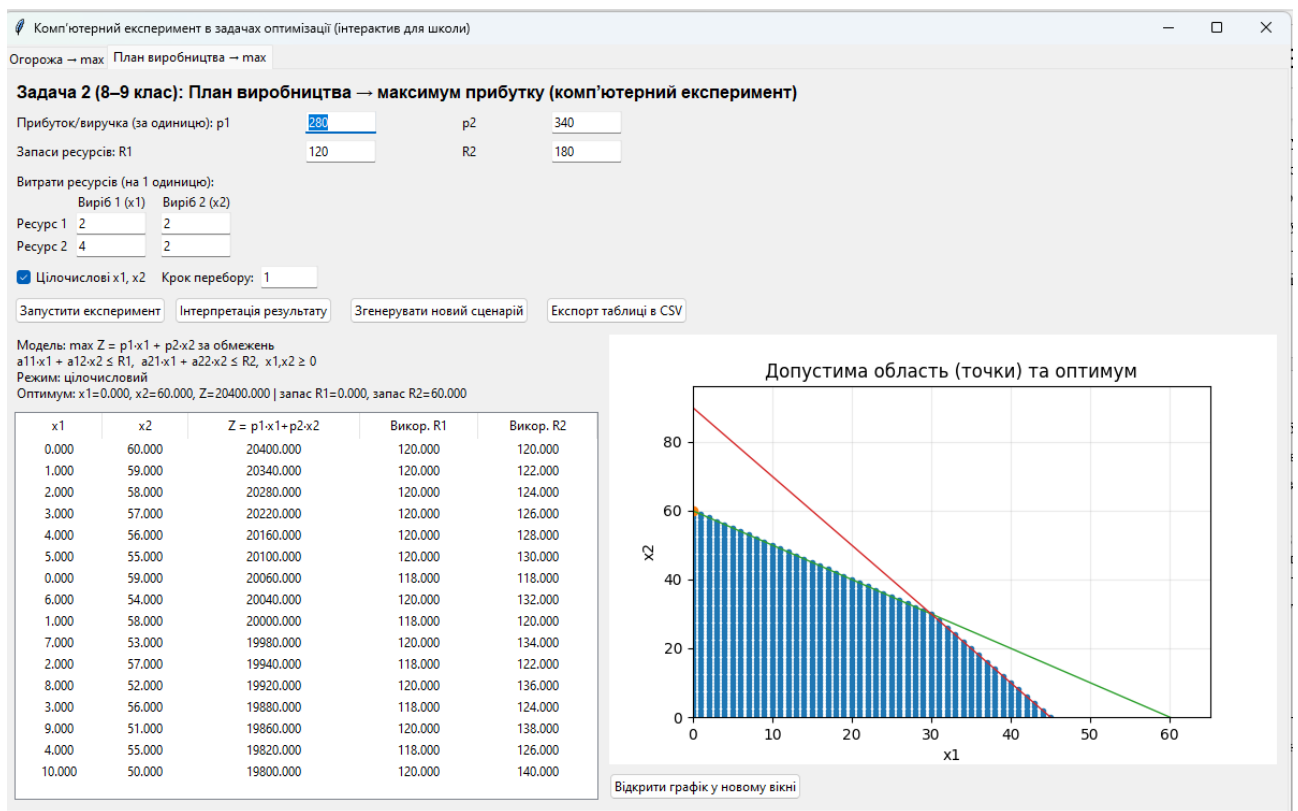


Рис. 2.4 – Вкладка “План виробництва → max”: область допустимих рішень (точки), граничні прямі обмежень та точка оптимуму

У другій вкладці також реалізовано окреме вікно інтерпретації, яке пояснює знайдений оптимум через поняття “активних обмежень” та “вузького місця”. У програмі це подається доступно: обчислюється, скільки ресурсу реально використано в оптимумі, і який запас ресурсу залишається, а потім

формується висновок, яке обмеження фактично стримує збільшення Z . Додатково в інтерпретації наводиться проста евристика “результат на одиницю ресурсу”, яка не підміняє оптимізацію, але допомагає учням змістовно пояснювати, чому в оптимальному плані може зростати частка одного виробу порівняно з іншим. У педагогічному сенсі це формує важливий місток між формальними нерівностями та реальними управлінськими висновками, що відповідає меті навчання моделюванню як засобу розуміння й прийняття рішень.

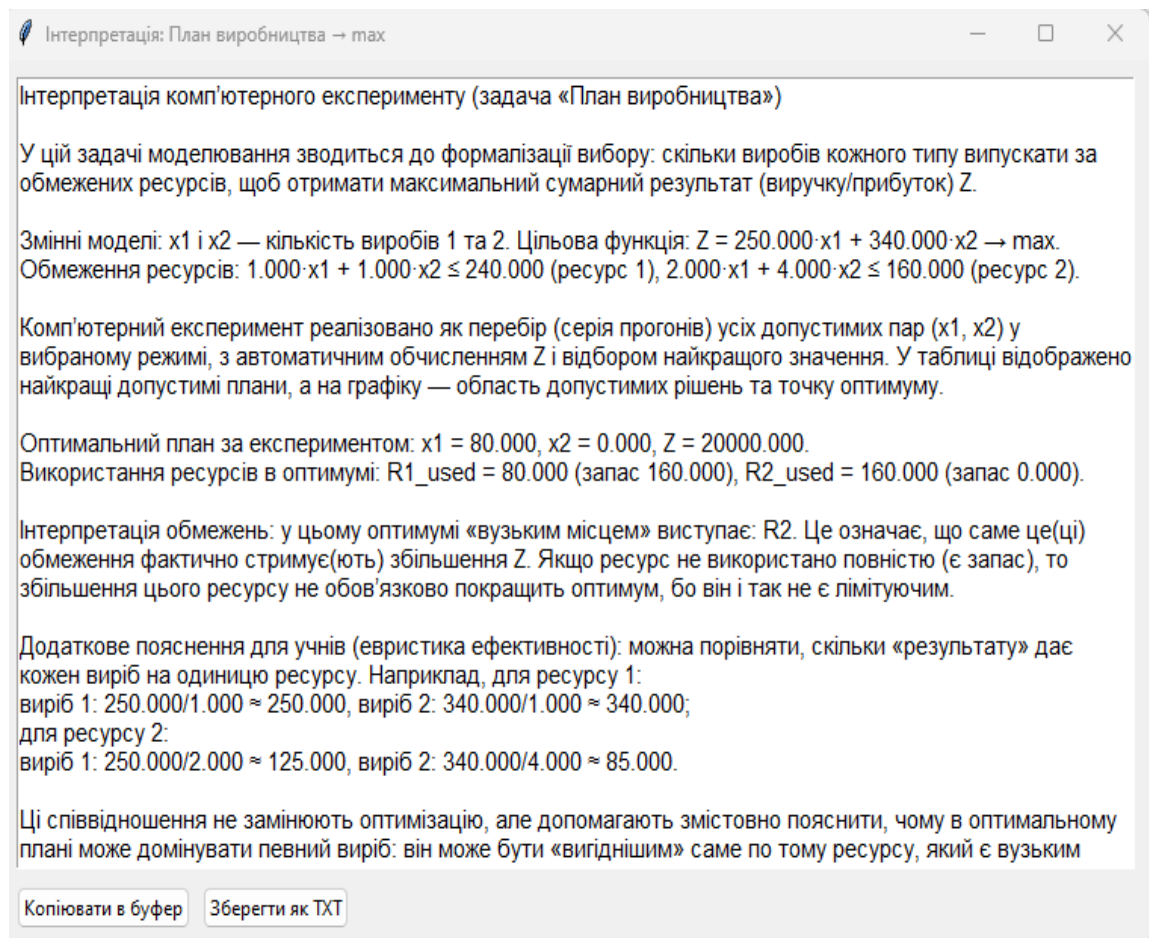


Рис. 2.5 – Вікно “Інтерпретація результатів” для задачі “План виробництва”: оптимум, використання ресурсів, активні обмеження та змістовний висновок

Для узагальнення функцій застосунок та їх дидактичного призначення доцільно зафіксувати відповідність між модульними компонентами програми й очікуваними результатами навчання. Така відповідність важлива, оскільки дозволяє обґрунтувати застосунок не як “демонстрацію”, а як інструмент

організації навчальної діяльності, у якій комп'ютерний експеримент виступає методом навчання.

Таблиця 2.1 – Функціональні модулі застосунку та їх дидактичне призначення в курсах математики й інформатики

Модуль застосунку	Вхідні параметри, що керуються учнем	Результат подання	Дидактичний ефект у навчанні моделювання та оптимізації
Параметричний прогін моделі “Огорожа”	(L) , крок дискретизації (b)	Таблиця $((b,a,S))$, графік $(S(b))$	Усвідомлення цільової функції, зв'язку між змінними та максимумом, поява поняття похибки чисельного експерименту
Параметричний прогін моделі “План виробництва”	$(p_1, p_2), (R_1, R_2)$, матриця витрат (a_{ij}) , p сітка	Таблиця допустимих планів і значень (Z) , графік області та оптимуму	Розуміння ролі обмежень, пошук найкращого плану як задачі оптимізації, інтерпретація “вузького місця”

Вікно інтерпретації результатів	Поточний знайдений оптимум і похідні показники	Текстове пояснення результату	Перехід від “числа” до змістовного висновку, формування навичок аргументації та рефлексії над моделлю
Експорт у CSV	Дані останнього експерименту	Файл CSV для Excel/Google Sheets	Розвиток культури роботи з даними, перенесення результатів у табличні процесори для подальшого аналізу

Архітектурно застосунок можна подати як взаємодію чотирьох логічних блоків: інтерфейс керування параметрами, обчислювальне ядро експерименту, модуль візуалізації та модуль інтерпретації/експорту. Така схема корисна для уроків інформатики, коли вчитель пояснює, як поділяються відповідальності між компонентами програми й чому це підвищує надійність та керованість розробки.



Рис 2.6 Логічна схема роботи застосунку

Практична цінність застосунку для уроків інформатики полягає в тому, що він є завершеним прикладом прикладного програмного засобу, де поєднуються подійно-орієнтований інтерфейс, чисельні обчислення та візуалізація. Учитель може демонструвати, як параметри з GUI перетворюються на дані експерименту, як результати структуруються для таблиці, як будується графік у вікні застосунку, і як забезпечується відтворюваність через експорт у CSV.

A1						
x1						
	A	B	C	D	E	F
1	x1		Z	R1_used	R2_used	
2	80.000000	0.000000	20000.000	80.000000	160.000000	
3	78.000000	1.000000	19840.000	79.000000	160.000000	
4	79.000000	0.000000	19750.000	79.000000	158.000000	
5	76.000000	2.000000	19680.000	78.000000	160.000000	
6	77.000000	1.000000	19590.000	78.000000	158.000000	
7	74.000000	3.000000	19520.000	77.000000	160.000000	
8	78.000000	0.000000	19500.000	78.000000	156.000000	
9	75.000000	2.000000	19430.000	77.000000	158.000000	
10	72.000000	4.000000	19360.000	76.000000	160.000000	
11	76.000000	1.000000	19340.000	77.000000	156.000000	
12	73.000000	3.000000	19270.000	76.000000	158.000000	
13	77.000000	0.000000	19250.000	77.000000	154.000000	
14	70.000000	5.000000	19200.000	75.000000	160.000000	
15	74.000000	2.000000	19180.000	76.000000	156.000000	
16	71.000000	4.000000	19110.000	75.000000	158.000000	
17	75.000000	1.000000	19090.000	76.000000	154.000000	
18	68.000000	6.000000	19040.000	74.000000	160.000000	
19	72.000000	3.000000	19020.000	75.000000	156.000000	
20	76.000000	0.000000	19000.000	76.000000	152.000000	
21	69.000000	5.000000	18950.000	74.000000	158.000000	
22	73.000000	2.000000	18930.000	75.000000	154.000000	
23	66.000000	7.000000	18880.000	73.000000	160.000000	
24	70.000000	4.000000	18860.000	74.000000	156.000000	
25	74.000000	1.000000	18840.000	75.000000	152.000000	
26	67.000000	6.000000	18790.000	73.000000	158.000000	
27	71.000000	3.000000	18770.000	74.000000	154.000000	
28	75.000000	0.000000	18750.000	75.000000	150.000000	
29	64.000000	8.000000	18720.000	72.000000	160.000000	
30	68.000000	5.000000	18700.000	73.000000	156.000000	
31	72.000000	2.000000	18680.000	74.000000	152.000000	
32	65.000000	7.000000	18630.000	72.000000	158.000000	
33	69.000000	4.000000	18610.000	73.000000	154.000000	
34	73.000000	1.000000	18590.000	74.000000	150.000000	
35	62.000000	9.000000	18560.000	71.000000	160.000000	

Рис 2.7 – Результати в CSV файлі

Для уроків математики ключовим є інше: застосунок дозволяє вивести учня на рівень дослідження моделі, коли він висуває припущення (наприклад, як зміниться оптимум, якщо збільшити (L) або ресурс (R₁)), перевіряє його серією прогонів і формулює змістовний висновок у вікні інтерпретації. Така практика посилює математичне мислення, оскільки акцент переноситься з одиничного обчислення на аналіз залежностей та обґрунтування рішення, а технологія стає інструментом перетворення “статичної” задачі на “динамічний” об’єкт

дослідження, що відповідає сучасним підходам інтеграції технологій у математичне навчання [NCTM 2024] та компетентнісним орієнтирам розвитку обчислювального мислення [OECD 2023].

Нижче наведено фрагмент коду, який демонструє сутність реалізації комп'ютерного експерименту: параметричний прогін, пошук максимуму, формування таблиці та побудову графіка, а також механізм відкриття вікна інтерпретації. Повний лістинг програми доцільно винести в додатки, тоді як у тексті достатньо показати саме “ядро”, яке відповідає темі комп'ютерного експерименту в задачах оптимізації.

Лістинг 2.1 – Фрагмент реалізації комп'ютерного експерименту та інтерпретації (Python)

```
def compute(self, L: float, step: float):
    if step <= 0:
        step = 1.0
    b_vals = np.arange(step, L / 2, step)
    a_vals = L - 2 * b_vals
    S_vals = a_vals * b_vals
    return b_vals, a_vals, S_vals

def run_experiment(self):
    L = float(self.L_var.get())
    step = safe_float(self.step_var.get(), 1.0)
    b, a, S = self.compute(L, step)
    idx = int(np.argmax(S)) # критерій оптимуму (максимум)
    b_star, a_star, S_star = b[idx], a[idx], S[idx]
    self.info.config(
        text=f"Експеримент: b*={b_star:.3f}, a*={a_star:.3f}, S*={S_star:.3f}"
    )
    self.plot.clear()
    self.plot.ax.grid(True, alpha=0.25)
    self.plot.ax.plot(b, S, lw=2)
    self.plot.ax.plot([b_star], [S_star], marker="o")
    self.plot.draw()

def show_interpretation(self):
    if self._interp_win is None or not self._interp_win.wininfo_exists():
        self._interp_win = InterpretationWindow(self, "Інтерпретація")
    self._interp_win.set_text(self.build_interpretation_text())
    self._interp_win.lift()
```

2.3. Методика впровадження задач оптимізації у процес навчання моделюванню

Методика впровадження задач оптимізації у навчання моделюванню в закладах загальної середньої освіти доцільно будувати як педагогічно керований цикл комп'ютерного експерименту, у якому оптимізація виступає не “окремою темою”, а способом дослідження моделі через зміну параметрів, порівняння альтернатив і тлумачення отриманого оптимуму в контексті реальної ситуації. Такий підхід узгоджується з сучасним розумінням математичної грамотності як здатності формулювати, застосовувати й інтерпретувати математику в різних контекстах, де моделювальний цикл і робота з даними є центральними процесами навчальної діяльності.

У межах цієї роботи методика спирається на три взаємопов'язані принципи. Перший полягає в тому, що комп'ютерний експеримент розглядається як систематична процедура, яка робить модель “спостережуваною”: учень керує входами (параметрами, кроком дискретизації, діапазонами змінних), а далі аналізує виходи (таблицю результатів, графік поведінки цільової функції, повідомлення про дотримання обмежень) і формулює висновок щодо найкращого варіанта. Саме такий дослідницький формат комп'ютерного математичного експерименту описується як продуктивний для шкільного навчання, зокрема завдяки візуалізації, доступності динамічних середовищ і можливості “наближати” абстрактні конструкції до практики.

Другий принцип пов'язаний із поетапним нарощуванням складності моделей і форм оптимізації. На початковому рівні (6–7 класи) оптимізація вводиться як пошук максимуму або мінімуму функції однієї змінної на обмеженій множині, де головним результатом є розуміння, що “найкраще” значення виникає як закономірний наслідок зв'язків між величинами, а не як випадковий підбір. На середньому рівні (8–9 класи) оптимізація розширюється до задач із двома змінними та ресурсними обмеженнями, де учні починають розуміти роль допустимої області, поняття лімітуючих (активних) обмежень і

сенс оптимуму як найкращого плану за умов дефіциту ресурсів. Важливо, що такий перехід здійснюється не лише за рахунок ускладнення формул, а через зміни способу експериментування: від одномірного параметричного прогону до перебору сітки, від одного графіка функції до графіка області допустимих рішень і ранжування множини варіантів у таблиці.

Третій принцип стосується дидактичної “повноти” комп’ютерного експерименту. Методика вважається реалізованою лише тоді, коли учень проходить шлях від постановки проблеми до інтерпретації результату і перевірки адекватності моделі, оскільки саме інтерпретація формує зв’язок між математичною формою й реальним змістом. У цьому сенсі технології мають не замінювати мислення, а посилювати його через репрезентації, дискурс і контрольовані експериментальні запуски, що співзвучно сучасним позиціям щодо відповідального використання інтелектуальних цифрових інструментів у математичному навчанні.

Запропонована методика організується як типова навчальна траєкторія в межах теми “Моделювання” (інформатика) або “Прикладні задачі/функції/нерівності” (математика), але ключовою одиницею планування є не параграф підручника, а навчальне завдання, яке має реалістичний контекст, модельну структуру та “точку оптимізації”, що може бути досягнута через комп’ютерний експеримент. У термінах педагогічного дизайну доцільно виділяти три рівні навчального результату. Перший рівень відображає предметні знання: учень розуміє змінні, параметри, обмеження, цільову функцію і відмінність між допустимими та недопустимими рішеннями. Другий рівень описує моделювальну компетентність: учень уміє переходити від тексту ситуації до математичної моделі, пояснювати припущення, виконувати перевірку узгодженості й робити висновок у контексті задачі. Третій рівень відповідає обчислювальній складовій: учень здатний організувати експеримент (вибір діапазонів, кроку, кількості прогонів), інтерпретувати табличні та графічні репрезентації, фіксувати результат і пояснювати похибку дискретного пошуку. Така трирівнева постановка корелює з рамками, у яких математичне

модельовання розглядають як цикл процесів і практик, а не як набір окремих правил .

Операційно методика впровадження оптимізаційних задач у навчання модельованню реалізується через сценарій заняття, що має стійку логіку, але допускає гнучкість у виборі інструмента. На першій фазі вчитель організує “проблематизацію” контексту, де оптимальність має змістовне пояснення: максимум площі за фіксованого периметра, мінімум витрат за заданого обсягу, максимум результату за дефіциту ресурсів, найкращий прогноз за критерієм похибки. У цей момент важливо домогтися, щоб учні сформулювали запитання у вигляді “що саме ми оптимізуємо?” і “які умови обмежують вибір?”, оскільки ці дві репліки фактично є проєкцією цільової функції та системи обмежень. На другій фазі здійснюється формалізація: вводяться змінні та параметри, фіксуються одиниці виміру, уточнюються допустимі значення, визначається математичний критерій оптимальності. На третій фазі учні планують комп’ютерний експеримент: обирають діапазон змін змінних і крок дискретизації, визначають кількість прогонів, задають спосіб контролю обмежень і формат фіксації результатів у таблиці. На четвертій фазі відбувається виконання експерименту в програмі або в електронних таблицях з отриманням набору результатів, який можна ранжувати. На п’ятій фазі здійснюється аналіз і інтерпретація: учні пояснюють, чому саме знайдений варіант є оптимальним, чи узгоджується він із графіком/таблицею, які обмеження виявилися критичними і як зміниться оптимум при модифікації параметрів. На завершенні вчитель організує коротку валідацію моделі через перевірку здорового глузду, порівняння з альтернативними кроками дискретизації або повторний запуск із зміненими даними, що формує уявлення про надійність і відтворюваність експерименту як ознаку наукового підходу.

Для забезпечення технологічної й методичної готовності доцільно комбінувати два класи інструментів. Перший клас становлять спеціалізовані навчальні застосунки для комп’ютерного експерименту (у роботі це інтерактивний застосунок із вкладками задач оптимізації, табличним виводом і

графіками), які зменшують “технічну ціну входу” та дозволяють концентрувати увагу на моделі, параметрах і сенсі оптимуму. Другий клас складають універсальні інструменти, передусім електронні таблиці, які є доступними в школі й природно підтримують табличні обчислення, побудову графіків, аналіз залежностей і прості прогнози. Саме поєднання спеціалізованого засобу й табличного процесора створює методичну гнучкість: учень бачить той самий експеримент у різних репрезентаціях, а вчитель може адаптувати інструмент до ресурсу кабінету та рівня класу. Ідея використання систем комп’ютерного моделювання як середовищ для проєктування пізнавальних завдань, у яких цифровий інструмент підтримує структурування діяльності учня, детально розкривається в українських дослідженнях цифрових освітніх середовищ [Биков та ін., 2020].

Ключовим методичним питанням є проєктування самих задач оптимізації як навчальних об’єктів. У запропонованій методиці задача має бути побудована так, щоб учень міг однозначно ідентифікувати модельні елементи: змінні як керовані величини, параметри як задані умови, обмеження як “фільтр” допустимих варіантів, критерій як міра успіху, а також інтерпретацію як пояснення того, що означає оптимум. У задачах для 6–7 класів домінує зв’язок “змінна $\rightarrow$ результат”, тому оптимальність має бути видимою через графік і таблицю; найкраще працюють сюжети з геометричним змістом або з простими залежностями, де цільова функція має зрозумілий сенс. У задачах для 8–9 класів додається конкуренція ресурсів і компроміс: покращення одного показника потребує більшого ресурсу або зменшує можливості іншої змінної, тому обмеження мають бути інтерпретовані як реальні “стелі” або “ліміти”. Тоді комп’ютерний експеримент перестає бути “перебиранням цифр” і стає дослідженням структури: учень може показати, що оптимум лежить у зоні, де принаймні одне обмеження є активним, і саме це пояснює, чому подальше збільшення цільової функції неможливе без зміни умов. Ця логіка безпосередньо підтримує перехід від інтуїтивного розуміння оптимальності до формування моделювальних умінь, що лежать у центрі рамок математичної грамотності.

Методика також передбачає керовану роботу з дискретизацією як педагогічним інструментом. У шкільній практиці оптимізація часто асоціюється з аналітичним знаходженням екстремумів, однак у моделюванні важливо показати, що комп'ютерний експеримент зазвичай працює з дискретним набором варіантів, особливо коли змінні є цілочисловими або коли модель має обмеження, які складно розв'язати “вручну”. Тому доцільно навчати учнів обґрунтовано обирати крок дискретизації, розуміти компроміс між точністю та часом обчислення і пояснювати похибку як наслідок дискретного пошуку. У практичному занятті це реалізується через подвійний прогін: спочатку з грубим кроком, щоб швидко побачити загальну форму залежності та область оптимуму, потім з дрібнішим кроком, щоб уточнити результат і порівняти зміни. Важливо, щоб вчитель підкреслював: у комп'ютерному експерименті “точність” є керованим параметром, а процедура уточнення є типовою науковою практикою, яка пов'язує результат з методикою його отримання. Такий формат добре узгоджується з ідеєю комп'ютерного експерименту як дидактичного методу, що формує уявлення про дослідницьку природу математики та інформатики [Shabat, 2023].

У контексті інтеграції з інформатикою методика передбачає, що учні повинні усвідомити відмінність між моделлю і програмною реалізацією моделі. На рівні навчальних дій це досягається через те, що математичні елементи (змінні, цільова функція, обмеження) фіксуються в тексті задачі та в зошиті, а програмний інструмент розглядається як спосіб організації експерименту та репрезентації результатів. У роботі із застосунком це проявляється в тому, що учень вводить параметри, запускає експеримент, отримує таблицю та графік, після чого переносить найкращий варіант у математичний запис і дає змістовне пояснення. У роботі з Excel учень формує таблицю прогонів, застосовує формули, використовує діаграму та інструменти тренду/регресії або підбору параметра залежно від задачі; однак принциповим залишається те, що інструмент “не вигадує” модель, а лише обчислює її наслідки. Такий наголос відповідає сучасним вимогам до використання цифрових і, ширше, інтелектуальних

інструментів у математичному навчанні, де підкреслюється відповідальність учителя й учня за зміст, коректність і інтерпретацію результату [NCTM, 2024].

Окремим елементом методики є організація інтерпретації результатів як формалізованої навчальної дії. У практиці комп'ютерного експерименту типова помилка полягає в тому, що учні сприймають оптимум як “число на екрані”, не пов'язуючи його з моделлю. Тому в методиці інтерпретація задається як обов'язковий результат: учень має пояснити, що саме оптимізується, чому знайдений варіант допустимий, які обмеження визначають форму допустимої області, як зміна параметрів впливатиме на оптимум і які висновки можна зробити для реальної ситуації. У застосунку цей крок підтримується вікном інтерпретації, але методично важливо, щоб текст інтерпретації став матеріалом для обговорення: учитель може запропонувати учням зіставити інтерпретацію з графіком, знайти в таблиці сусідні варіанти й пояснити, чому вони гірші, або змінити параметр і перевірити, чи зберігається попередній висновок. Така організація розвиває мовлення в математиці, аргументацію та рефлексію над моделлю, що відповідає ключовим орієнтирам моделювального циклу в міжнародних рамках оцінювання і навчання [OECD, 2023].

Для контролю якості впровадження методика має містити прозорий підхід до оцінювання, який фіксує не лише правильність числового результату, а й коректність моделі та обґрунтованість висновку. З огляду на це доцільно використовувати критерії, що відображають повний цикл діяльності: адекватність формалізації (правильність змінних і обмежень), коректність експериментальної постановки (діапазони, крок, відтворюваність), якість аналізу репрезентацій (таблиця/графік) та змістовність інтерпретації. У практиці це дозволяє оцінити прогрес учня навіть у випадку, коли оптимум “майже правильний”, але помітна похибка кроку або некоректно обраний діапазон значень; тоді учень отримує не штраф, а навчальне завдання з уточнення експерименту, що підтримує дослідницьку логіку. Підґрунтям для такого підходу слугує ідея, що навчання моделюванню є навчанням процесу мислення і

перевірки, а не лише підстановки у формули, що послідовно відображено в сучасних аналітичних рамках математичної грамотності.

Таблиця 2.2 – Узагальнена карта методики впровадження оптимізаційних задач у навчання моделюванню

Фаза навчальної роботи	Зміст моделювальної дії	Цифровий інструмент як підтримка	Очікуваний навчальний продукт
Постановка задачі в контексті	Виокремлення того, що саме вважається “кращим”, і що обмежує вибір	Візуальний вступ, демонстраційні дані, приклад таблиці	Формулювання критерію та умов у природній мові
Формалізація моделі	Опис змінних, параметрів, системи обмежень, цільової функції	Ввід параметрів у застосунок або структурування в Excel	Математичний запис моделі з поясненням одиниць
Планування експерименту	Вибір діапазонів, кроку, режиму перебору, фіксація структури даних	Налаштування кроку/сітки, вибір режиму у вкладці, підготовка шаблону таблиці	План експерименту, що забезпечує відтворюваність
Проведення експерименту	Серія прогонів, відсів недопустимих варіантів, пошук оптимуму	Автоматичне формування таблиці і графіка, експорт у CSV	Результативна таблиця, графік, знайдений оптимум
Аналіз, інтерпретація, валідація	Пояснення оптимуму, активних обмежень, похибки,	Вікно інтерпретації, повторні запуски зі зміненими параметрами	Змістовний висновок і коротке обґрунтування на основі репрезентацій

	перевірка узгодженості		
--	---------------------------	--	--

Методика не може бути повноцінною без уваги до організаційних умов впровадження в школі: часу уроку, різного рівня підготовки учнів, обмежень технічної бази, а також педагогічної безпеки при використанні цифрових інструментів. У практичному плані найбільш життєздатним є формат, коли комп'ютерний експеримент використовується як ядро одного або двох уроків у межах теми моделювання, а результати оформлюються в короткий звіт, який містить модель, таблицю експерименту, графік і інтерпретацію. Якщо ресурс дозволяє, методика підсилюється мікропроектом на 2–3 тижні, де учні змінюють контекст, але зберігають метод: обирають власну ситуацію, формалізують, планують експеримент, виконують серію запусків, роблять висновок і демонструють, як зміна параметрів змінює оптимум. Саме така “стабільність методу при змінності контекстів” є показником того, що учні засвоїли оптимізацію як інструмент моделювання, а не як одиничний тип задачі. У дослідженнях про комп'ютерний математичний експеримент підкреслюється, що найбільший освітній ефект виникає тоді, коли цифровий інструмент слугує середовищем для систематичного дослідження, а не тільки для демонстрації готових відповідей, і коли комп'ютерне моделювання розглядається як спосіб проектування пізнавальних завдань у цифровому освітньому середовищі.

Щоб методика була технологічно прозорою для учнів і водночас зручною для вчителя, доцільно фіксувати логіку “моделювання–експеримент–інтерпретація” у вигляді схеми, яку можна включати в робочі матеріали уроку, а також використовувати як основу для самооцінювання учня після виконання завдання. Нижче наведено варіант такої схеми, який відображає не лише послідовність дій, а й ідею повернення до моделі у випадку, якщо експеримент виявив суперечність, надмірну похибку або неузгодженість з контекстом.



Рис 2.8 Методичний цикл впровадження оптимізації у навчання

У підсумку запропонована методика впровадження задач оптимізації у процес навчання моделюванню є цілісною системою, де оптимізація виступає не лише математичною процедурою, а дидактичним механізмом формування дослідницьких умінь через комп'ютерний експеримент, таблично-графічні репрезентації та обов'язкову інтерпретацію результатів. Вона дозволяє органічно поєднати зміст математики та інформатики, забезпечити кероване ускладнення від задач з однією змінною до моделей із ресурсними обмеженнями, а також сформувати в учнів практику перевірки й уточнення моделі як ознаку наукового мислення, що відповідає сучасним міжнародним рамкам математичного

моделювання та рекомендаціям щодо відповідального використання цифрових інструментів у навчанні.

ВИСНОВОК ДО РОЗДІЛУ 2

У розділі 2 узагальнено й апробовано практичні підходи до організації комп'ютерного експерименту в задачах оптимізації під час вивчення моделювання у закладах загальної середньої освіти. Показано, що навчальні задачі оптимізаційного змісту для 6–9 класів можуть бути методично коректно подані як послідовний цикл “контекст → формалізація → план експерименту → обчислювальний прогін → таблично-графічний аналіз → інтерпретація результату”, де цифровий інструмент не замінює міркування учня, а забезпечує кероване варіювання параметрів і прозору фіксацію результатів, що підсилює дослідницький характер навчання.

Практична реалізація комп'ютерного експерименту на прикладі інтерактивного застосунку продемонструвала дидактичну доцільність поєднання кількох репрезентацій результату — таблиці значень, графіка поведінки цільової функції або області допустимих рішень, а також текстового модуля інтерпретації. Така зв'язка забезпечує перехід від отримання “числа оптимуму” до змістовного пояснення, чому саме цей варіант є найкращим за заданими обмеженнями, які параметри впливають на оптимум та як зміниться результат при модифікації умов. Доведено, що керована дискретизація (вибір кроку, діапазону та режиму перебору) може бути використана як навчальний інструмент для формування уявлень про точність, похибку та відтворюваність експерименту, що є важливим елементом наукового стилю мислення у моделюванні.

Запропонована методика впровадження оптимізаційних задач у навчання моделюванню забезпечує логічне нарощування складності: від задач з однією змінною та наочною графічною інтерпретацією до задач із кількома змінними й ресурсними обмеженнями, у яких оптимум пояснюється через “лімітуючі” умови та структуру допустимої області. Показано, що інтеграція математики й

інформатики в цьому підході є змістовною: математика відповідає за постановку й обґрунтування моделі (змінні, обмеження, критерій), тоді як інформатика забезпечує організацію експерименту (алгоритм перебору, подання даних, візуалізація, експорт результатів), що підвищує практичну цінність навчання та мотивацію учнів за рахунок інтерактивності.

Отже, результати розділу 2 підтверджують, що комп'ютерний експеримент у задачах оптимізації є ефективним інструментом навчання моделюванню в школі за умови методично правильної організації: обов'язкової інтерпретації результатів, використання таблично-графічних репрезентацій, контролю точності через дискретизацію та опори на реалістичні контексти задач, які роблять критерій "оптимальності" змістовно зрозумілим для учнів.

ВИСНОВКИ

У висновках до роботи узагальнено теоретичні, методичні та прикладні результати дослідження, присвяченого використанню комп'ютерного експерименту в задачах оптимізації під час вивчення моделювання в закладах загальної середньої освіти. У ході виконання роботи уточнено зміст поняття комп'ютерного експерименту як інструмента дослідження математичної моделі шляхом керованого варіювання параметрів і аналізу таблично-графічних результатів, а також обґрунтовано його дидактичну цінність як способу переведення навчання моделюванню з репродуктивного рівня на дослідницький. Показано, що оптимізаційні задачі є методично продуктивним ядром такого підходу, оскільки природно вимагають чіткого визначення змінних, обмежень і критерію, а отже, формують системне розуміння моделі та відповідальність за інтерпретацію отриманого оптимуму.

Теоретичний блок роботи дав змогу систематизувати базові підходи до математичного моделювання та оптимізації, визначити роль цифрових інструментів у підтримці моделювального циклу й розкрити особливості комп'ютерного експерименту як методу навчання, що поєднує обчислення, візуалізацію, перевірку гіпотез та рефлексивне тлумачення результатів. Обґрунтовано, що застосування комп'ютерного експерименту є найбільш ефективним тоді, коли технологія не підміняє змістовні міркування, а забезпечує прозоре відтворюване середовище для серії запусків моделі з фіксацією наслідків у різних репрезентаціях. Це дозволяє учням спостерігати залежності між параметрами та результатами, порівнювати альтернативні рішення та робити висновки на основі даних, що безпосередньо підсилює формування моделювальної компетентності.

Практична частина підтвердила придатність запропонованого підходу для рівня 6–9 класів і продемонструвала, що навчальні задачі оптимізації можуть бути організовані як єдиний методичний цикл “контекст → формалізація → план

експерименту → обчислювальний прогін → таблично-графічний аналіз → інтерпретація”. Апробація показала, що поєднання таблиці результатів, графічного подання та окремого модуля інтерпретації забезпечує перехід від отримання “числа оптимуму” до пояснення його змісту й причинно-наслідкових зв’язків у моделі, зокрема через аналіз впливу дискретизації, діапазону перебору та ресурсних обмежень. Встановлено, що керована дискретизація може виступати важливим дидактичним інструментом: учні отримують можливість усвідомити компроміс між точністю та обчислювальною складністю, а також сформувати уявлення про похибку як властивість процедури експериментального пошуку.

Методика впровадження задач оптимізації в процес навчання моделюванню, запропонована у роботі, забезпечує логічне нарощування складності від задач з однією змінною та наочною залежністю до задач із двома змінними та системою обмежень, де оптимальне рішення пояснюється через структуру допустимої області та “лімітуючі” ресурси. Показано, що такий підхід створює змістовну інтеграцію математики та інформатики: математика забезпечує постановку моделі й обґрунтування критеріїв, а інформатика — організацію комп’ютерного експерименту, структурування даних, візуалізацію та відтворюваність результатів. У підсумку доведено, що комп’ютерний експеримент у задачах оптимізації є ефективним засобом підвищення мотивації та якості засвоєння моделювання за рахунок інтерактивності, наочності та можливості самостійного дослідження моделі учнями.

Отже, поставлену мету роботи досягнуто: теоретично обґрунтовано та методично конкретизовано використання комп’ютерного експерименту в задачах оптимізації при вивченні моделювання в загальній середній освіті, а також показано його прикладну реалізацію через навчальні завдання і цифрові інструменти. Практична значущість результатів полягає в можливості безпосереднього використання розробленої методики та матеріалів на уроках математики й інформатики, а також у придатності підходу до подальшого розширення на інші контексти оптимізації та більш складні моделі за умови

збереження головного дидактичного принципу: результат комп'ютерного експерименту має завершуватися аргументованою інтерпретацією та перевіркою адекватності моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Балик Н. Р. Вибрані питання комп'ютерного моделювання процесів і систем. Тернопіль : ТНПУ, 2022. 160 с.
2. Балик Н. Р., Гарах О. А. Комп'ютерне моделювання в STEM освіті. Тернопіль : ТНПУ, 2023. 120 с.
3. Биков В. Ю., Спирін О. М., Пінчук О. П. Сучасні інформаційні технології та інноваційні методики навчання. Київ : Педагогічна думка, 2017. 310 с.
4. Борисенко В. Д., Устенко С. А., Устенко І. В. Основи комп'ютерного моделювання в інженерній діяльності : навч. посіб. Миколаїв : МНУ, 2016. 276 с.
5. Бурда М. І., Тарасенкова Н. А. Математика. Алгебра та початки аналізу. Підручник. Київ : Ранок, 2018. 288 с.
6. Вовк Н. Комп'ютерне моделювання електротехнічних комплексів та систем. Наукові записки ТНПУ. Серія : Технічні науки. 2015. № 2. С. 45–52.
7. Горбатюк Р. М. Комп'ютерне моделювання у підготовці майбутніх інженерів-педагогів до професійної діяльності. Тернопіль : ТНПУ, 2009. 180 с.
8. Горошкіна О. М. Розвиток логічного мислення учнів засобами комп'ютерного моделювання. Педагогічні науки. 2018. № 78. С. 124–130.
9. Горошко Ю. В. Використання Excel у розв'язуванні прикладних задач з математики. Математика в школах України. 2019. № 19. С. 12–17.
10. Дронь В. В. Використання комп'ютерного моделювання для педагогічних цілей. Тернопіль : ТНПУ, 2017. 98 с.
11. Жалдак М. І., Рашевська Н. В. Комп'ютерно-орієнтовані методичні системи навчання математики. Київ : НПУ ім. М. П. Драгоманова, 2011. 264 с.
12. Hillier F. S., Lieberman G. J. *Introduction to Operations Research*. – 11th ed. – Dubuque : McGraw-Hill Education, 2021.

13. Boyd S. P., Vandenberghe L. *Convex Optimization*. – Cambridge : Cambridge University Press, 2004. – 716 p.
14. Winston W. L. *Operations Research: Applications and Algorithms*. – 4th ed. – Belmont, CA : Thomson/Brooks/Cole, 2004. – 1418 p.
15. Новікова А. О. Формування в учнів основної школи умінь математичного моделювання у процесі навчання алгебри : дис. ... канд. пед. наук : 13.00.02. – Київ, 2021. – 305 с.
16. Цифрова трансформація освіти: концептуальні засади та методичні аспекти : монографія. – Київ : Вид-во УДУ імені М. П. Драгоманова, 2024. – 382 с. – ISBN 978-966-931-312-6.
17. Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестопапов Є. А. Інформатика : підручник для 9 класу закладів загальної середньої освіти. – Харків : Ранок, 2019. – 180 с
18. Жалдак М. І., Триус Ю. В. Основи теорії і методів оптимізації : навч. посіб. Черкаси : Брама-Україна, 2005. 608 с.
19. Заблоцький А. Шляхи оптимізації професійної підготовки майбутніх працівників соціальної сфери. Тернопіль : ТНПУ, 2020. 140 с.
20. Козак Л. В., Жосан Г. В. Використання цифрових технологій у процесі формування математичних компетентностей. Інформаційні технології і засоби навчання. 2021. Т. 82, № 2. С. 45–59.
21. Крамаренко Т. Г., Семеніхіна О. В. ГеоГебра в навчанні математики: методичний посібник. Суми : СумДПУ ім. А. С. Макаренка, 2012. 112 с.
22. Литвинова С. Г. Хмаро орієнтоване навчальне середовище. Київ : ІТЗН, 2016. 348 с.
23. Ліштаба Т. В. Оптимізація навчального процесу як компонент моделі педагогічної діяльності. Тернопіль : ТНПУ, 2021. 152 с.
24. Ліштаба Т. В. Теоретико-методична модель педагогічної діяльності в контексті моделювання та оптимізації. Тернопіль : ТНПУ, 2021. 164 с.

25. Ліштаба Т. В. Функціональна модель діяльності інженера-педагога комп'ютерного профілю. Вісник Національної академії державної прикордонної служби. 2018. № 3. С. 55–62.
26. Луцик І. І. Використання онтологій для програмних модулів адаптивних систем керування на базі нечіткої логіки. Тернопіль : ТНПУ, 2023. 132 с.
27. Мартинюк С. В., Вишневський С. Оптимізаційні модельні підходи. Сучасні інформаційні технології : зб. тез доп. 2021. С. 16.
28. Муранова Н. П. Оптимізація моделювання освітніх процесів. Тернопіль : ТНПУ, 2014. 128 с.
29. Оліяр О. Створення системи візуалізації та моделювання бездротових сенсорних мереж. Тернопіль : ТНПУ, 2015. 112 с.
30. Пантелят М., Кузьмін А. Програмні засоби для комп'ютерного моделювання мультифізичних процесів у електромагнітах. Вісник НТУ «ХПІ». 2022. № 2(8). С. 15–19. DOI: <https://doi.org/10.20998/2079-3944.2022.2.03>
31. Пасічник О. Комп'ютерне моделювання об'єктів і процесів. Комп'ютерний експеримент. 10 клас. URL: <https://dystosvita.org.ua/mod/page/view.php?id=592>
32. Пастовенський О. В. Структурна модель оптимізації регіональних освітніх мереж. Рідна школа. 2008. № 6. С. 30–33.
33. Постумент М. В., Грод І. М. Комп'ютерне моделювання як інноваційний метод навчання. Тернопіль : ТНПУ, 2017. 36 с.
34. Проектування комп'ютерних моделей систем : електронний ресурс. URL: https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t1/11..htm
35. Романишина О. Я. Оптимізація об'єкта віртуального середовища. Тези ІТ конференції ТНПУ. 2022.
36. Романишина О. Я. Програмні засоби для розробки 3D-моделі віртуального туру та оптимізація. Сучасні інформаційні технології : тези конф. 2020.

37. Савченко І. М. Комп'ютерні технології навчання математики: теорія і практика. Київ : Київський ун-т ім. Б. Грінченка, 2019. 226 с.
38. Семеніхіна О. В., Дерев'янка Д. П. GeoGebra як засіб формування математичних компетентностей школярів. Інформаційні технології і засоби навчання. 2016. Т. 52, № 2. С. 31–43.
39. Тафінцева С. М. Моделювання педагогічних ситуацій у підготовці майбутнього вчителя початкової школи. Тернопіль : ТНПУ, 2009. 140 с.
40. Теплицький І. О. Моделювання та оптимізація в задачах з курсу математики. Математика в школі. 2017. № 4. С. 15–19.
41. Триус Ю. В. Комп'ютерно-орієнтовані системи навчання. Черкаси : Брама, 2000. 384 с.
42. Федорейко В. С., Горбатюк Р. М., Ільків І. Імітаційне комп'ютерне моделювання як ефективний засіб підготовки майбутніх учителів. Тернопіль : ТНПУ, 2019. 148 с.
43. Федчишин О. М., Мохун І. Методичні особливості застосування комп'ютерного моделювання. Наукові праці ТНПУ. Серія : Педагогіка. 2018. № 2. С. 152–159.
44. Хворостіна Ю. В., Юрченко А. О. Особливості навчання комп'ютерному моделюванню на уроках інформатики. Науковий вісник Ужгородського національного університету. Серія : Педагогіка. Соціальна робота. 2022. Вип. 1(50). С. 333–336.
45. Чайка В. М., Петрова І. М. Інтелектуальна культура вчителя як фактор інновацій у педагогічних процесах. Тернопіль : ТНПУ, 2014. 120 с.
46. Швець В. О. STEM-освіта в Україні: теорія та практика впровадження. Київ : Педагогічна думка, 2020. 256 с.

ДОДАТКИ

Додаток А. План-конспект практичної роботи для 6 класу

Тема: Комп'ютерний експеримент у задачі оптимізації: максимальна площа ділянки за огорожею на три сторони.

Тип уроку: комбінований (коротке введення + практична робота з моделлю).

Тривалість: 45 хв.

Обладнання та ПЗ: ПК/ноутбук (1 на 1–2 учні), Excel або Google Sheets, мультимедійний екран (за наявності).

Мета: сформувати уявлення про оптимізацію як пошук найкращого варіанта за умов обмеженого ресурсу; навчити виконувати комп'ютерний експеримент у табличному процесорі на основі моделі з однією керуючою змінною.

Очікувані результати: учень(учениця) коректно вводить змінні та обмеження, будує табличну модель, проводить обчислювальний прогін із заданим кроком, візуалізує залежність графіком, визначає максимум і формулює змістовний висновок.

Навчальні передумови (що учні вже мають знати).

Учні оперують поняттями прямокутника, площі та периметра, виконують обчислення у таблиці, розуміють, що формула у клітинці може посилатися на інші клітинки.

Ключові поняття уроку (у викладі вчителя в тексті, без переліків). Упродовж уроку вводяться й уточнюються поняття “модель”, “керуюча змінна”, “обмеження ресурсу”, “цільова функція” та “комп'ютерний експеримент як серія розрахунків при зміні параметра з фіксацією результатів у таблиці й на графіку”.

Логіка проведення уроку

У вступній частині вчитель подає сюжет: потрібно обгородити прямокутну ділянку біля природної межі (річка, стіна, схил), тому огорожа потрібна лише на трьох сторонах. Учитель акцентує, що реальна задача має два виміри: з одного

боку є ресурсне обмеження (довжина огорожі), з іншого — бажаний результат (максимальна площа ділянки). На цьому етапі вчитель організовує коротке обговорення, щоб учні самостійно сформулювали, що означає “найкращий” варіант у цій ситуації: це той, що дає найбільшу площу, не виходячи за межі наявної огорожі.

Далі вчитель разом з учнями переходить від сюжету до математичної моделі. Позначається ширина ділянки b як сторона, яка огорожується двічі, та довжина a як сторона, яка огорожується один раз. Ресурсне обмеження задається рівнянням $L = a + 2b$, звідки $a = L - 2b$. Площа S визначається формулою $S = ab$, а після підстановки залежності $a = L - 2b$ отримуємо $S(b) = b(L - 2b)$. Учитель пояснює, що в цій моделі b є керуючою змінною: ми будемо змінювати її значення, а a та S автоматично перераховуватимуться. Так організується комп’ютерний експеримент: серія прогонів для різних b з метою знайти максимум площі.

Практична частина реалізується у Excel/Sheets як відтворюваний експеримент. Учитель задає конкретне значення L для класного прогону (наприклад, $L = 60$ м або інше значення, яке вчитель вважає зручним), наголошуючи, що це лише приклад, а модель працює для будь-якого L . Учні створюють таблицю з трьома основними стовпцями: значення b , розрахунок a за формулою $a = L - 2b$ і розрахунок площі $S = a \cdot b$. Для забезпечення методичної точності вчитель проговорює, що допустимі значення b мають бути такими, щоб $a > 0$, тобто $L - 2b > 0$, а також $b > 0$. У таблиці це забезпечується вибором діапазону b , наприклад від 1 до $L/2 - 1$, щоб уникнути нульової довжини. Учитель демонструє перший рядок таблиці на екрані та показує, як правильно “протягнути” формули вниз, щоби таблиця автоматично заповнилась.

Коли таблицю заповнено, учні будують графік залежності площі S від ширини b . Учитель пояснює, що графік у цьому завданні є не “ілюстрацією”, а способом перевірки результату: максимум площі має проявитися як вершина кривої, а табличний максимум має відповідати найбільшій точці на графіку. Далі

організовується коротке уточнення результату через ідею дискретизації: учитель пропонує знайти максимум при кроці 1, а потім — у вузькому діапазоні біля знайденого максимуму виконати додатковий прогін із кроком 0,5 (або 0,2), щоб продемонструвати, що менший крок дає точніший результат. На цьому етапі учні порівнюють два результати й формулюють висновок, що комп'ютерний експеримент має параметр точності (крок), який впливає на наближеність знайденого оптимуму.

Підсумкова інтерпретація оформлюється як короткий науково коректний висновок: учень фіксує оптимальні значення b і a , значення максимальної площі $S_{\max}$, а також пояснює, яким способом це підтверджено (таблиця значень і графік). Учитель звертає увагу, що коректний висновок має містити не лише числа, а й пояснення причинно-наслідкового зв'язку “зміна $b \rightarrow$ зміна $a \rightarrow$ зміна S ” та обґрунтування, що обмеження на огорожу не порушено.

Оцінювання результату практичної роботи (у формі критеріїв у тексті).

Робота оцінюється за трьома компонентами: правильність математичної моделі (коректно записано обмеження $L = a + 2b$ та формули $a = L - 2b$, $S = a \cdot b$); коректність комп'ютерної реалізації (таблиця заповнена формулами, без ручних підстановок, побудовано графік залежності); якість інтерпретації (учень робить висновок про оптимальні розміри й пояснює, як саме за таблицею й графіком встановлено максимум, а також зазначає роль кроку дискретизації в точності).

Домашнє завдання (методичне продовження експерименту).

Учневі пропонується взяти інше значення L (наприклад, змінити ресурс огорожі на 80 м або 100 м) і повторити експеримент у двох режимах точності (крок 1 і крок 0,5), після чого коротко порівняти, чи змінився оптимальний характер рішення та наскільки відрізняються значення $S_{\max}$.

Додаток Б. План-конспект практичної роботи для 7 класу (деталізовано)

Тема: Комп'ютерний експеримент у задачі оптимізації: максимальна площа прямокутника, вписаного в коло радіуса r (переріз колоди).

Тип уроку: урок-практикум (математичне моделювання + графічний аналіз комп'ютерний експеримент).

Тривалість:

45

хв.

Обладнання та ПЗ: Excel/Google Sheets; за можливості демонстраційно GeoGebra (для візуалізації “рухомої” геометрії).

Мета: сформувати навичку переходу від геометричної ситуації до формульного обмеження та виконати оптимізаційний пошук через параметричний комп'ютерний експеримент.

Очікувані результати: учень(учениця) встановлює обмеження $a^2 + b^2 = 4r^2$, реалізує модель у таблиці, будує графік $S(a)$, знаходить максимум у допустимій області та обґрунтовує оптимальний переріз у змістовних термінах.

Навчальні передумови та методична ідея

Практична робота передбачає, що учні вже вміють обчислювати площу прямокутника, застосовувати теорему Піфагора до прямокутного трикутника, а також вводити формули в електронні таблиці та будувати діаграми. Методична ідея полягає в тому, що оптимізація подається як дослідження функції площі на допустимій множині, а комп'ютерний експеримент виступає як технологічний спосіб виконати серію обчислень і візуально встановити максимум.

Логіка проведення уроку

1. Мотиваційно-вступний етап (приблизно 5 хв)

Учитель задає прикладний контекст: уявімо круглу колоду з поперечним перерізом у вигляді кола радіуса r . Потрібно вирізати балку прямокутного перерізу так, щоб площа перерізу була максимальною, адже це означає найбільший “корисний” об'єм деревини при фіксованому діаметрі колоди. Учитель уточнює, що “знайти максимум” означає вибрати такі сторони

прямокутника, щоб він залишався вписаним у коло і водночас мав найбільшу площу.

2. Формалізація: геометрія $\rightarrow$ модель (приблизно 10 хв)

Учні разом з учителем фіксують геометричну основу моделі. Прямокутник вписаний у коло, а його діагональ проходить через центр кола і дорівнює діаметру $2r$. Якщо сторони прямокутника позначити a і b , то діагональ є гіпотенузою прямокутного трикутника з катетами a і b . Отже, за теоремою Піфагора отримуємо обмеження:

$$a^2 + b^2 = (2r)^2 = 4r^2.$$

Учитель підкреслює, що це — **умова допустимості**: будь-які a і b , які не задовольняють цю залежність, описують прямокутник, що не може бути вписаним у коло заданого радіуса.

Цільова величина (цільова функція) визначається як площа прямокутника:

$$S = ab.$$

Далі учитель вводить ідею параметризації: щоб провести комп'ютерний експеримент, зручно зробити керуючою змінною одну зі сторін, наприклад a . Тоді b виражається з обмеження:

$$b = \sqrt{4r^2 - a^2},$$

а площа стає функцією однієї змінної:

$$S(a) = a \cdot \sqrt{4r^2 - a^2}.$$

Учитель окремо акцентує, що допустимі значення a мають виконувати умову $4r^2 - a^2 \geq 0$, тобто $0 \leq a \leq 2r$. Це принципово важливо для коректної побудови експерименту, адже при виході за ці межі підкореневий вираз стає від'ємним і модель втрачає зміст.

3. Побудова комп'ютерного експерименту в Excel/Sheets (приблизно 18–20 хв)

3.1. Підготовка блоку параметрів

Учитель пропонує для класного прогону конкретне значення радіуса, наприклад $r = 20\text{см}$ (або інше значення, зручне для обчислень). Учні створюють у верхній частині аркуша клітинку “ r ” та вводять значення 20. Винесення r в окрему клітинку робить модель гнучкою: при зміні r автоматично перераховується вся таблиця та графік.

3.2. Структура таблиці та заповнення керуючої змінної

У першому стовпці учні створюють колонку значень a . Учитель пояснює, що комп'ютерний експеримент є дискретним: ми не перебираємо всі можливі значення безперервно, а задаємо крок. Для першого прогону доцільно взяти крок 1 см (або 0,5 см, якщо дозволяє час), задавши значення a від 0 до $2r$. Важливо, щоб учні бачили вплив кроку: більший крок дає швидке, але грубіше наближення до максимуму, менший — точніше, але повільніше.

3.3. Обчислення b через обмеження

У другому стовпці розраховується b за формулою $b = \sqrt{4r^2 - a^2}$. Учитель показує, як правильно організувати посилання на клітинку з r , щоб при копіюванні формули вниз вона не “з’їхала” на інші клітинки. У Excel це забезпечується абсолютним посиланням на клітинку r .

Типовий вигляд формули (залежить від адрес клітинок; наведено як шаблон):

=SQRT(4*\$B\$1^2 - A4^2)

де \$B\$1 — клітинка з r , A4 — поточне значення a .

У Google Sheets аналогічно, з тим самим принципом.

Учитель додає контроль коректності: якщо a випадково вийде за межі $2r$, підкореневий вираз стане від’ємним, і таблиця видасть помилку. Це має трактуватися не як “погана програма”, а як сигнал порушення допустимої області моделі. За потреби вчитель демонструє, як уникнути помилок, обмеживши

діапазон a або використавши логічну перевірку (наприклад, IF), але наголошує, що методично правильніше — **не задавати недопустимі a** .

3.4. Обчислення площі та пошук максимуму

У третьому стовпці учні обчислюють площу $S = a \cdot b$. Формула має посилатися на поточні значення a і b у тому самому рядку.

Після заповнення таблиці учні знаходять максимальне значення площі. Учитель пропонує два способи контролю: по-перше, за допомогою функції MAX() для стовпця площі; по-друге, “візуально-таблично”, порівнюючи значення в таблиці поблизу найбільшого. Це дозволяє зробити висновок, що максимум визначається і чисельно, і графічно.

3.5. Побудова графіка $S(a)$ і графічна інтерпретація

Учні виділяють стовпці a і S та будують графік залежності S від a (лінійна або точкова діаграма). Учитель пояснює, що графік є ключовим елементом комп’ютерного експерименту: він дає змогу швидко побачити, в якій зоні функція зростає, де досягає вершини і де спадає. На графіку максимум відповідає найвищій точці/вершині кривої.

4. Уточнення максимуму через зміну кроку (сценарний міні-експеримент) (приблизно 5–6 хв)

Щоб підкреслити дослідницький характер комп’ютерного експерименту, учитель пропонує уточнення: після того як грубий прогін з кроком 1 см показав область максимуму, учні звужують діапазон a навколо знайденого максимуму і виконують повторний прогін з меншим кроком (наприклад 0,2 або 0,5 см). Учитель пояснює, що це відповідає підходу “спочатку грубе наближення, потім локальне уточнення”, який широко застосовується в чисельних методах та оптимізації.

5. Підсумкова інтерпретація результатів (приблизно 4–5 хв)

На підсумковому етапі учні формулюють висновок у логіці “таблиця → графік → пояснення”. У висновку обов’язково зазначається, які значення a і b відповідають максимальній площі і як це підтверджено двома способами:

числово (максимум у таблиці) та візуально (вершина кривої на графіку). Учитель додатково звертає увагу на якісне пояснення: оптимальний переріз має властивість “збалансованості”, тобто при максимальній площі сторони прямокутника наближаються одна до одної за величиною. У межах моделі це означає, що надмірне збільшення a призводить до суттєвого зменшення b через обмеження $a^2 + b^2 = 4r^2$, і площа втрачається, тому максимум досягається при компромісі.

Оцінювання

Оцінювання здійснюється за сукупністю компонентів. Робота вважається виконаною коректно, якщо учень(учениця) правильно сформулював(ла) геометричне обмеження і коректно відобразив(ла) його у вигляді формули для b , побудував(ла) таблицю значень для допустимого діапазону a , використав(ла) правильні посилання на параметр r та забезпечив(ла) коректне копіювання формул. Високий рівень передбачає наявність графіка $S(a)$, правильне встановлення точки максимуму та змістовну інтерпретацію результату із поясненням ролі обмеження й допустимої області. Окремо враховується вміння учня пояснити, чому недопустимі значення a призводять до втрати геометричного змісту, і як це контролюється під час планування експерименту.

Домашнє завдання

Учню пропонується змінити параметр r (наприклад, взяти $r = 15\text{см}$ або $r = 25\text{см}$), повторити комп’ютерний експеримент і порівняти максимальну площу та оптимальні значення a і b у двох випадках. У висновку учень має пояснити, як зміна радіуса впливає на значення максимального перерізу, спираючись на таблицю та графік, а не на припущення.

Додаток В. План-конспект практичної роботи для 8 класу

Тема: Цілочислова оптимізація як комп'ютерний експеримент: мінімізація кількості рейсів у задачі перевезення двох видів вантажу (Excel Solver).

Тип уроку: практична робота з комп'ютерним експериментом і оптимізаційним інструментом.

Тривалість: 45 хв.

Обладнання та ПЗ: Excel з активованою надбудовою Solver; робота в парах.

Мета: навчити формалізувати прикладну задачу як оптимізаційну модель з обмеженнями та цілочисловими змінними і розв'язувати її засобами Solver.

Очікувані результати: учень(учениця) задає змінні, цільову функцію і обмеження, коректно налаштовує Solver, отримує оптимальний план і пояснює його.

Логіка проведення уроку

Учитель пропонує задачу: необхідно перевезти задані обсяги вантажу двох типів, використовуючи кілька “типів рейсів”, де кожен рейс має фіксовану комбінацію вантажів. Учні вводять змінні x_1, x_2, x_3 як кількість рейсів кожного типу. Цільова функція — мінімізувати загальну кількість рейсів $Z = x_1 + x_2 + x_3$. Обмеження формулюються як вимоги виконати перевезення кожного виду вантажу не менше заданого, а також невід'ємність і цілочисловість x_i .

У таблиці Excel учні будують структуру моделі так, щоб усі розрахунки були прозорими: окремо виділяються клітинки зі змінними x_i , окремо — матриця “скільки перевозиться за один рейс”, далі — клітинки сумарно перевезених обсягів як лінійні комбінації, і окремо — клітинка цільової функції Z . Після цього вчитель покроково демонструє налаштування Solver: вибір клітинки Z як Objective у режимі Min, вибір клітинок x_i як Changing Variable Cells, додавання обмежень на обсяги перевезень і додавання вимоги integer для змінних. Учитель

наголошує, що саме цілочисловість відрізняє цю постановку від звичайного “підбору” і наближає її до реального планування.

Після отримання розв’язку учні виконують верифікацію: перевіряють, чи задовольняються обмеження, та інтерпретують, що означають значення x_i у сюжеті. Далі вводиться короткий сценарний експеримент: змінюється один параметр (наприклад, необхідний обсяг вантажу S), повторюється Solver і порівнюється, як змінився оптимальний план. Це дозволяє показати, що комп’ютерний експеримент включає не лише пошук одного розв’язку, а й дослідження чутливості оптимуму до умов.

Оцінювання базується на коректності постановки моделі в таблиці, правильності налаштувань Solver та якості інтерпретації результату.

Додаток Г. План-конспект практичної роботи для 9 класу

Тема: Прогнозування та оптимізаційний підбір параметрів у комп'ютерному експерименті: дискретна модель «дві популяції хижаків – одна жертва» (Excel).

Тип уроку: інтегрований практикум (моделювання + таблиця + графіки + підбір параметра).

Тривалість: 45 хв.

Обладнання та ПЗ: Excel / Google Sheets; бажано Excel із функцією “Підбір параметра” (Goal Seek) або надбудовою Solver.

Мета: навчити реалізовувати рекурентну модель у таблиці, будувати прогноз на 10 років, аналізувати графіки динаміки, а також виконувати оптимізаційний підбір одного параметра для досягнення заданого критерію стійкості системи.

Очікувані результати: учень(учениця) будує дискретну модель з рекурентними формулами, отримує прогноз, інтерпретує динаміку системи за графіками, проводить сценарний аналіз і демонструє керований підбір параметра як елемент оптимізації в комп'ютерному експерименті.

Навчальні передумови та міжпредметні зв'язки

Зміст практикуму спирається на навички роботи з електронними таблицями (введення даних, формули, відносні/абсолютні посилання, копіювання формул, побудова діаграм) та на уявлення про моделювання як формалізований опис реального процесу. Міжпредметний характер роботи проявляється у поєднанні математичного апарату (функціональні залежності, рекурентні співвідношення, аналіз графіків) з елементами інформатики (структурування даних, обчислювальний експеримент, налаштування інструментів підбору) та природничих наук (інтерпретація популяційної динаміки як екологічного процесу).

Логіка проведення уроку (детально)

1. Організаційно-мотиваційний етап (приблизно 5 хв)

Учитель задає реалістичний екологічний сюжет: у певній місцевості є популяція **жертви** (наприклад, миші), а також дві популяції **хижаків** (лисиці та сови), що живляться цією жертвою. Потрібно не “вгадати майбутнє”, а **побудувати модель** і за допомогою комп’ютера **дослідити**, як змінюватиметься система при заданих параметрах. Учитель підкреслює, що комп’ютерний експеримент у цій роботі має дві частини: по-перше, **прогнозування** (обчислення траєкторій за роками), по-друге, **оптимізаційний етап** (керований підбір одного параметра, щоб виконати цільову умову стійкості).

Для методичної чіткості учитель озвучує, що учні працюватимуть у двох режимах: базовий сценарій (за початковими параметрами) та керований сценарій (після підбору параметра), після чого порівнюють результати.

2. Формалізація: від сюжету до дискретної моделі (приблизно 8–10 хв)

Учитель разом з учнями вводить позначення: N_t — чисельність жертви у році t , F_t — чисельність першого хижака (лисиці), O_t — чисельність другого хижака (сови). Кожна чисельність описується по роках, тому прогноз задається рекурентно, тобто через перехід від t до $t + 1$.

У поданій навчальній версії моделі зміни чисельності жертви трактуються як сумарний ефект природного приросту і втрат від смертності та поїдання. Тому базовий перехід задається співвідношенням

$$N_{t+1} = N_t + g_N \cdot N_t - d_N \cdot N_t - c_F \cdot F_t - c_O \cdot O_t,$$

де g_N — коефіцієнт приросту жертви (за рік), d_N — коефіцієнт смертності жертви (за рік), c_F та c_O — середня кількість жертви, яку знищує один хижак відповідного виду за рік.

Для хижаків у спрощеному шкільному варіанті (щоб фокус зберігався на рекурентності, таблиці та підборі параметрів) використовується модель “приріст–смертність”:

$$F_{t+1} = F_t + g_F \cdot F_t - d_F \cdot F_t, O_{t+1} = O_t + g_O \cdot O_t - d_O \cdot O_t,$$

де g_F, g_O — коефіцієнти приросту, а d_F, d_O — коефіцієнти смертності популяцій хижаків. Учитель окремо пояснює, що це — **модельне припущення**: у реальності популяції хижаків також залежать від доступності їжі, однак у рамках даного практикуму використовується спрощення, достатнє для демонстрації логіки комп'ютерного експерименту та оптимізаційного підбору параметрів.

Оскільки чисельність популяцій не може бути від'ємною, вводиться обмеження типу “нижня межа 0”. У таблиці це реалізується функцією $\text{MAX}(0; \dots)$.

3. Комп'ютерна реалізація моделі в Excel (приблизно 15 хв)

Учні відкривають Excel і створюють файл з двома логічними блоками: **блок параметрів і розрахункову таблицю за роками**.

У блоці параметрів (у верхній частині аркуша) учні вводять початкові значення N_0, F_0, O_0 , коефіцієнти приросту та смертності $g_N, d_N, g_F, d_F, g_O, d_O$, а також норми споживання c_F і c_O . Учитель наголошує, що винесення параметрів в окремі клітинки є ключовою вимогою до якісної моделі: тоді будь-який сценарний експеримент зводиться до зміни одного параметра без переписування формул.

Далі створюється розрахункова таблиця з рядками років від $t = 0$ до $t = 10$ та стовпцями “Рік”, “Жертва N_t ”, “Хижак 1 F_t ”, “Хижак 2 O_t ”. У рядку $t = 0$ залишаються посилання на параметричні клітинки N_0, F_0, O_0 . Починаючи з рядка $t = 1$, вводяться формули рекурентного переходу, які посилаються на значення попереднього року.

Наведений нижче приклад показує принцип (точні адреси клітинок залежать від макета учня; важливо, щоб параметри були зафіксовані абсолютними посиланнями). Для україномовних/європейських налаштувань Excel часто використовується роздільник $;$, для англійськомовних — $;$; учитель дозволяє обидва варіанти, якщо формула працює коректно.

Для жертви в рядку року $t + 1$ використовується конструкція: $=\text{MAX}(0; N_t + \$gN\$N_t - \$dN\$N_t - \$cF\$F_t - \$cO\$O_t)$

Для хижака 1: $=\text{MAX}(0; F_t + \$gF\$F_t - \$dF\$F_t)$

Для хижака 2: $=\text{MAX}(0; O_t + \$gO\$O_t - \$dO\$O_t)$

Після введення формул у рядок $t = 1$ учні копіюють формули до рядка $t = 10$. Учитель контролює, щоб параметри посилалися на одну й ту саму клітинку (абсолютні посилання), а змінні N_t, F_t, O_t були відносними посиланнями на попередній рядок.

Після цього учні будують діаграму динаміки популяцій. Оптимальною для навчальних цілей є лінійна діаграма з трьома рядами даних (жертва і два хижаки) на одному полі або дві діаграми: одна для N_t , інша — для F_t і O_t . Учитель пояснює, що графік виконує функцію “візуального тесту”: якщо модель або формули помилкові, графік часто демонструє неприродні стрибки, різкі провали або некоректні значення.

4. Інтерпретація результатів базового сценарію (приблизно 5–7 хв)

Учні аналізують отримані траєкторії та дають змістовну інтерпретацію. В інтерпретації вчитель вимагає, щоб учень(учениця) відповів(ла) не лише “що сталося”, а й “чому саме так” у межах моделі. Для цього учні пов’язують тенденції з параметрами: зростання N_t пояснюється домінуванням додатного приросту над смертністю та споживанням, спад — навпаки; стабілізація або коливання пояснюються тим, що за певного співвідношення коефіцієнтів втрати і приріст частково компенсують одне одного. Якщо трапляється різке падіння чисельності жертви до нуля, учитель трактує це як сигнал про “нестійкий режим” у моделі (у реальності — ризик зникнення популяції), і підводить до потреби керованого параметричного втручання.

У цей момент учитель вводить поняття **критерію стійкості** як формальної умови, яку можна перевірити в таблиці. Наприклад, встановлюється вимога: чисельність жертви на 10-му році має бути не меншою за порогове значення $N_{\min}$ (його задає вчитель, наприклад 3000 або інше значення, придатне до

конкретних параметрів моделі). Порогове значення трактують як мінімально прийнятний рівень популяції для “стабільного існування системи” в межах навчальної інтерпретації.

5. Оптимізаційний підбір параметра: Goal Seek / Solver (приблизно 8–10 хв)

Оптимізаційний компонент реалізується як керований підбір **одного** параметра, що відповідає ідеї “керуючої змінної” в задачах оптимізації, але вже в контексті динамічної моделі. Учитель пояснює, що тепер у нас є ціль у вигляді обмеження на результат прогнозу: потрібно забезпечити $N_{10} \geq N_{\min}$. У ролі керуючого параметра обирається коефіцієнт, який піддається регулюванню в сюжеті. Найбільш методично прозорий варіант — підбирати **коефіцієнт смертності одного з хижаків** d_F або d_O , що може інтерпретуватися як зменшення чисельності хижака через природні причини/міграцію, або підбирати **коефіцієнт приросту жертви** g_N , що може трактуватися як поліпшення умов існування жертви.

Якщо використовується **Goal Seek**, учні виконують такі дії: обирають клітинку з N_{10} (чисельність жертви на 10-му році) як цільову, задають їй бажане значення $N_{\min}$ і вказують, яку клітинку параметра змінювати (наприклад, d_F). Після виконання процедури Excel знаходить таке значення параметра, за якого N_{10} досягає заданого порога. Учитель звертає увагу, що Goal Seek реалізує “підбір” під рівняння, тому його зручно застосовувати, коли контролюється одна цільова величина і змінюється один параметр.

Якщо використовується **Solver**, учитель може показати більш “оптимізаційну” постановку: мінімізувати зміну параметра від базового значення за умови $N_{10} \geq N_{\min}$. Наприклад, створюється допоміжна клітинка $\Delta = |d_F - d_{F0}|$ або $\Delta = (d_F - d_{F0})^2$, і Solver мінімізує Δ при обмеженні $N_{10} \geq N_{\min}$ та допустимих межах параметра (наприклад $0 \leq d_F \leq 0,5$). Така постановка демонструє, що оптимізація може включати не лише “досягти порогу”, а й “змінити систему мінімально необхідно”.

Після підбору параметра учні знову аналізують графік і порівнюють його з базовим сценарієм: у якій мірі підбір параметра змінив траєкторію жертви, чи зменшилися ризики падіння нижче порога, як змінилися популяції хижаків. На цьому етапі важливо зафіксувати методичний висновок: оптимізація у комп'ютерному експерименті — це кероване коригування моделі для досягнення заданого критерію, причому результат залежить від прийнятих припущень і параметрів.

Підсумковий контроль і оцінювання

Оцінювання практичної роботи здійснюється комплексно, оскільки результатом є не лише “таблиця з числами”, а повний цикл комп'ютерного експерименту. Високий рівень виконання передбачає, що учень(учениця) коректно виніс(ла) всі параметри в окремий блок, побудував(ла) таблицю років 0–10 і використав(ла) рекурентні формули з коректними посиланнями на попередній рік та на параметри. Обов'язковою умовою є наявність графіка(ів) і текстової інтерпретації, де пояснено тенденції та пов'язано їх із параметрами моделі. Окремо оцінюється оптимізаційний етап: учень має вміти пояснити, який саме параметр було обрано керуючим, яку цільову умову задано (критерій стійкості), яким інструментом здійснено підбір (Goal Seek чи Solver) і як перевірено, що умова виконана. Підсумковий висновок вважається коректним, якщо містить порівняння двох сценаріїв (базового та керованого), пояснює причини відмінностей і містить методичне узагальнення про роль комп'ютерного експерименту в задачах оптимізації при вивченні моделювання.

Домашнє завдання

Учневі пропонується змінити один параметр моделі (наприклад, норму споживання c_F або коефіцієнт смертності d_O) і повторити прогноз, порівнявши результат із базовим сценарієм. Далі учень виконує керований підбір іншого параметра для забезпечення того самого критерію стійкості $N_{10} \geq N_{\min}$ і формулює висновок, який підхід до регулювання виявився “ефективнішим” у

межах моделі (тобто який параметр довелося змінювати менше для досягнення порогу).