НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

*Кафедра автоматизованих систем обробки інформації та управління*

**Дипломний проєкт**

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_ *Олександр ПАВЛОВ*

(підпис) (вл.ім’я, прізвище)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2021 р.

**на здобуття ступеня бакалавра**

|  |
| --- |
| **за освітньо-професійною програмою «Інформаційні управляючі**  **системи та технології»** |
| **спеціальності 126 «Інформаційні системи та технології»** |

|  |
| --- |
| на тему: «*Назва…….* » |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Виконав (-ла):** | студент (-ка) *IV* курсу, групи *ІС-71* |  |  |
|  | *Іванов Петро Володимирович* |  |  |
|  | (прізвище, ім’я, по батькові) |  | (підпис) |
| **Керівник** | *доц., к.т.н., доц. Жданова Олена Григорівна* |  |  |
|  | (посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) |  | (підпис) |
| **Консультант з графічної документації** | *доц., к.т.н., доц. Тєлишева Тамара Олексіївна* |  |  |
|  | (посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) |  | (підпис) |
| **Рецензент** |  |  |  |
|  | (посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) |  | (підпис) |

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Київ – 2021

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

*інформатики та обчислювальної техніки*

Факультет (інститут)

(повна назва)

*автоматизованих систем обробки інформації та управління*

Кафедра

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_ *Олександр ПАВЛОВ*

(підпис) (вл.ім’я, прізвище)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2021 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

*Ситникову Владиславу Валерійовичу*

(прізвище, ім’я, по батькові)

|  |
| --- |
| 1. Тема проєкту «*Назва……* », |
|  |

керівник проєкту *Жданова Олена Григорівна, к.т.н., доцент я*

( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ *11* ” *05*  2021 р. № 1139-с

2. Термін подання студентом проєкту “*04*”*червня* 2021 року *я*

3. Вихідні дані до проєкту

|  |
| --- |
| *Технічне завдання* |
|  |

4. Зміст пояснювальної записки

|  |
| --- |
| *1. Загальні положення:* *основні визначення та терміни,* *опис предметного середовища,* *огляд ринку програмних продуктів,* *постановка задачі* |
| *2.* *Інформаційне забезпечення:* *вхідні дані,* *вихідні дані,* *опис структури бази даних* |
| *3. Математичне забезпечення: змістовна та математична постановки задачі,* |
| *обґрунтування та опис методу розв’язання* |
| *4.* *Програмне та технічне забезпечення**: засоби розробки,* *вимоги до* |
| *технічного забезпечення,* *архітектура програмного забезпечення,* *побудова звітів* |
| *5.* *Технологічний розділ: керівництво користувача, методика випробувань програмного продукту* |

5. Перелік графічного матеріалу

|  |
| --- |
| *1. Схема структурна варіантів використань* |
| *2. Схема структурна станів системи* |
| *3. Схема бази даних* |
| *4. Схема структурна класів програмного забезпечення* |
| *5. Рішення з математичного забезпечення* |
| *6. Креслення вигляду екранних форм* |

6. Консультанти розділів проєкту

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| завдання видав | завдання  прийняв |
|  |  |  |  |

7. Дата видачі завдання «*7*» *квітня 2021 року а*

Календарний план

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів виконання дипломного  проєкту | Термін виконання етапів проєкту | Примітка |
|  | *Вивчення рекомендованої літератури* |  |  |
|  | *Аналіз існуючих методів розв’язання задачі* |  |  |
|  | *Постановка та формалізація задачі* |  |  |
|  | *Розробка інформаційного забезпечення* |  |  |
|  | *Алгоритмізація задачі* |  |  |
|  | *Обґрунтування використовуваних технічних засобів* |  |  |
|  | *Розробка програмного забезпечення* |  |  |
|  | *Налагодження програми* |  |  |
|  | *Виконання графічних документів* |  |  |
|  | *Оформлення пояснювальної записки* |  |  |
|  | *Подання ДП на попередній захист* | *14.05.2021* |  |
|  | *Подання ДП на основний захист* | *04.06.2021* |  |
|  | *Подання ДП рецензенту* | *07.06.2021* |  |

Студент Владислав СИТНИКОВ

Керівник Олена ЖДАНОВА

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *№ з/п* | | *Формат* | *Позначення* | | | | | *Найменування* | | | | | *Кількість листів* | *Примітка* | |
| *1* | | *А4* |  | | | | | *Завдання на дипломний проєкт* | | | | | *2* |  | |
| *2* | | *А4* | *ДП 7101.00.000 ПЗ* | | | | | *Пояснювальна записка* | | | | | *67* |  | |
| *3* | | *А4* | *ДП 7101.01.000 ТЗ* | | | | | *Технічне завдання* | | | | | *15* |  | |
| *4* | | *А3* | *ДП 7101.02.000 ССВ* | | | | | *Схема структурна варіантів використань* | | | | | *1* |  | |
| *5* | | *А3* | *ДП 7101.03.000 ССС* | | | | | *Схема структурна станів системи* | | | | | *1* |  | |
| *6* | | *А3* | *ДП 7101.04.000 СБД* | | | | | *Схема бази даних* | | | | | *1* |  | |
| *7* | | *А3* | *ДП 7101.05.000 ССК* | | | | | *Схема структурна класів програмного* | | | | | *1* |  | |
|  | |  |  | | | | | *забезпечення* | | | | |  |  | |
| *9* | | *А3* |  | | | | | *Рішення з математичного забезпечення* | | | | | *1* |  | |
| *10* | | *А3* | *ДП 7101.06.000 КЕ* | | | | | *Креслення вигляду екранних форм* | | | | | *1* |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  | |  |  | | | | |  | | | | |  |  | |
|  |  | | |  |  |  | *ДП 7101 00.000.00* | | | | | | | | |
|  |  | | |  |  |  |
| ***Зм.*** | ***Арк.*** | | | ***ПІБ*** | ***Підп.*** | ***Дата*** |
| *Розробн.* | | | | ПІБ студента |  |  | *Відомість  дипломного проєкту* | | *Літ.* | | | *Лист* | | | *Листів* |
| *Керівн.* | | | | ПІБ керівникаДП |  |  |  |  |  | *1* | | | *1* |
| *Консульт.* | | | |  |  |  | *КПІ ім. Ігоря Сікорського*  *Каф. АСОІУ*  *Гр. ІС-71* | | | | | | |
| *Н/контр.* | | | | ПІБ н/контр |  |  |
| *В.о.зав.каф.* | | | | *Павлов О.А.* |  |  |

АНОТАЦІЯ

*Зм.*

*Арк.*

*Прізвище*

*Підпис*

*Дата*

*Лист*

2

*ДП 7101.00.000 ПЗ*

*Розроб.*

*Іванов І.І.*

*Перевірив.*

*.*

ПІБ керівникаДП

*Н. кон.*

ПІБ н/контр*.*

*Затв.*

ПІБ керівникаДП

Тема згідно з наказом

*Літ.*

*Листів*

–

*КПІ ім. Ігоря Сікорського*

*Каф. АСОІУ*

*Гр. ІС-71*

**Структура та обсяг роботи.** Пояснювальна записка дипломного проєкту складається з п’яти розділів, містить ?? рисунків, ?? таблиць, ?? додатків, ?? джерел.

Дипломний проєкт присвячений розробці комплексу задач … . Цілі та задачі розробки.

У розділі інформаційного забезпечення …

Розділ математичного забезпечення присвячений…

Програмне забезпечення …

У технологічному розділі …

|  |
| --- |
| ТУТ МАЄ БУТИ ПЕРЕЛІК КЛЮЧОВИХ СЛІВ: ВЕЛИКИМ ЛІТЕРАМИ, ЧЕРЕЗ КОМУ. |
|  |

ABSTRACT

*Той самий текст, що й в анотації, але іноземною мовою (англійська, німецька, НЕ російська)*

ЗМІСТ

[Вступ 7](#_Toc303777083)

[1 Загальні положення 7](#_Toc303777084)

[1.1 Опис предметного середовища 7](#_Toc303777085)

[1.1.1 Опис процесу діяльності 7](#_Toc303777086)

[1.1.2 Опис функціональної моделі 7](#_Toc303777087)

[1.2 Огляд наявних аналогів 7](#_Toc303777088)

[1.3 Постановка задачі 7](#_Toc303777089)

[1.3.1 Призначення розробки 7](#_Toc303777090)

[1.3.2 Цілі та задачі розробки 7](#_Toc303777091)

Висновок до розділу

[2 Інформаційне забезпечення 7](#_Toc303777092)

[2.1 Вхідні дані 7](#_Toc303777093)

[2.2 Вихідні дані 7](#_Toc303777094)

[2.3 Опис структури бази даних 7](#_Toc303777095)

[2.4 Структура масивів інформації 7](#_Toc303777096)

[Висновок до розділу 7](#_Toc303777102)

[3 Математичне забезпечення 7](#_Toc303777097)

[3.1 Змістовна постановка задачі 7](#_Toc303777098)

[3.2 Математична постановка задачі 7](#_Toc303777099)

[3.3 Обґрунтування методу розв’язання 7](#_Toc303777100)

[3.4 Опис методу розв’язання 7](#_Toc303777101)

[Висновок до розділу 7](#_Toc303777102)

[4 Програмне та технічне забезпечення 7](#_Toc303777103)

[4.1 Засоби розробки 7](#_Toc303777104)

[4.2 Вимоги до технічного забезпечення 7](#_Toc303777105)

[4.2.1 Загальні вимоги 7](#_Toc303777106)

[4.2.2 Опис локальної обчислювальної мережі 7](#_Toc303777107)

[4.3 Архітектура програмного забезпечення 7](#_Toc303777108)

[4.3.1 Діаграма класів 7](#_Toc303777109)

[4.3.2 Діаграма послідовності 7](#_Toc303777110)

[4.3.3 Діаграма компонентів 7](#_Toc303777111)

[4.3.4 Специфікація функцій 7](#_Toc303777112)

[4.4 Опис звітів 7](#_Toc303777113)

[Висновок до розділу 7](#_Toc303777102)

[5 Технологічний розділ 7](#_Toc303777114)

[5.1 Керівництво користувача 7](#_Toc303777115)

[5.2 Випробування програмного продукту 7](#_Toc303777116)

[5.2.1 Мета випробувань 7](#_Toc303777117)

[5.2.2 Загальні положення 7](#_Toc303777118)

[5.2.3 Результати випробувань 7](#_Toc303777119)

[Висновок до розділу 7](#_Toc303777102)

[загальні висновки 7](#_Toc303777123)

[Перелік посилань 7](#_Toc303777124)

[ДОДАТОК A Тексти програмного коду 7](#_Toc303777126)

Вступ

*Загальні відомості щодо історії виникнення та обґрунтування необхідності розробки.*

*За необхідністю навести визначення та терміни, що стосуються теми дипломного проєкту.*

*При описі стандартних термінів чи інших відомостей (не введених Вами) необхідно посилатися на їх джерела.*

Дипломний проєкт присвяченийрозробці комплексу задач …

**Практичне значення одержаних результатів.** Розроблено алгоритм розв’язання двокритеріальної транспортної задачі.

**Публікації.** Результати роботи були опубліковані у ? статтях у наукових збірниках ??? [1 – 2] та ?? тезах доповідей на науково-технічних конференціях [3 – 5].

# Загальні положення

## **Опис предметного середовища**

Дробово-лінійне програмування є більш узагальненим видом лінійного програмування.

Загальною задачею лінійного програмування називається задача, що має на меті визначення максимального (мінімального) значення функції [6]:



де - задані постійні величини.

При умовах:

 ,

 ,

 ,

де - задані постійні величини і .

В лінійному програмуванні цільова функція є лінійною, тоді як в дробово-лінійному цільова функція є відношенням двох лінійних функцій. Таким чином лінійна програма може бути розглянута як випадок дробово-лінійної програми де в знаменнику функція є сталою.

В данній роботі будемо розглядати таку цільову функцію задачі дробово-лінійного програмування в умовах невизначеності:

, ,

де - задані постійні величини.

Під невизначеністю в даному випадку розуміється існування  можливих значень , .

### **Опис процесу діяльності**

Процес дослідження задачі дробово-лінійного програмування в умовах невизначеності вручну є достатньо довгим та трудомістким, тому розроблений продукт, що є десктоп застосунком, призначений для спрощення цього процесу.

Дослідник у системі має змогу розв’язати задачу попередньо ввівши вхідні дані вручну або задавши налаштування для випадкової генерації умови. Крім цього є можливість проведення експериментів. Для цього попередньо потрібно обрати сценарій експерименту, вести умову задачі та додаткові налаштування пов’язані з обраним сценарієм. Після проведених досліджень дослідник може зберегти результати та переглянути їх пізніше.

### **Опис функціональної моделі**

У графічних матеріалах відображена функціональна модель у вигляді UML-діаграми варіантів використання, де описано яким чином актор взаємодіє із системою.

У даній системі актором виступає Дослідник. Опишемо його основні функції:

* введення умови індивідуальної задачі;
* розв’язання індивідуальної задачі;
* збереження умови індивідуальної задачі у файл з розширенням .csv;
* збереження розв’язку індивідуальної задачі у файл з розширенням .csv;
* введення налаштування для проведення експериментів;
* виконання експериментів;
* створення графіків для наглядного представлення результатів експерименту;
* збереження налаштування експериментів;
* збереження результатів експериментів у файл із розширенням .csv;
* перегляд розв’язків індивідуальної задачі;
* перегляд результатів експериментів;

## **Огляд наявних аналогів**

Так як не виявлено програмного забезпечення, що має інструменти саме для вирішення та дослідження задачі дробово-лінійного програмування в умовах невизначеності в межах даної теми доцільно аналізувати програмні засоби, що вирішують такі ж підзадачі, що і розроблювана система

**[Math.semestr](Https://math.semestr.ru/)**

Режим доступу - <https://math.semestr.ru/simplex/fractional-linear-programming.php>.

Math.semestr – це онлайн калькулятор, що дає змогу розв’язувати детерміновану задачу дробово-лінійного програмування. Крім цього, дане програмне забезпечення має досить широкий функціонал для розв’язку задач у таких направленнях: вища математика, аналітична геометрія, матричні перетворення, теорія вірогідності, економетріка, статистика, мережева модель, інформатика, математичні методи у психології, методи оптимізації, динамічне програмування, лінійне програмування, дослідження операцій, системи масового обслуговування, теорія ігр та теорія автоматчного управління.

**Google OR-Tools**

Режим доступу - https://developers.google.com/optimization.

OR-Tools - це пакет програмного забезпечення з відкритим кодом для оптимізації, призначений для вирішення найскладніших проблем у світі з маршрутизацією транспортних засобів, потоками, цілочисельним та лінійним програмуванням та програмуванням обмежень.

Змоделювавши свою проблему вибраною мовою програмування, ви можете використовувати будь-який із півтора десятка вирішувачів: комерційні вирішувачі, такі як Gurobi або CPLEX, або вирішувачі з відкритим кодом, такі як SCIP, GLPK або GLOP від Google та CP-SAT.

Лінійний оптимізатор Glop знаходить оптимальне значення лінійної цільової функції, враховуючи набір лінійних нерівностей як обмеження (наприклад, призначення людей на роботу або пошук найкращого розподілу набору ресурсів при мінімізації витрат).

OR-Tools написаний на C ++, але ви також можете використовувати його з Python, Java або C#.

**LINGO Software**

Режим доступу - https://www.lindo.com/.

LINGO - це комплексний інструмент, призначений для швидшого, простішого та ефективнішого побудови та вирішення моделей лінійної, нелінійної (опуклої та неопуклої / глобальної), квадратичної, квадратично обмеженої, напіввизначеної, стохастичної та цілочисельної оптимізації. LINGO пропонує повністю інтегрований пакет, який включає потужну мову для вираження моделей оптимізації, повнофункціональне середовище для побудови та редагування проблем, а також набір швидких вбудованих вирішувачів.

Graphical user interface, application

Description automatically generated

Рисунок 1 – Приклад інтерфейсу LINGO Software

## **Постановка задачі**

### **Призначення розробки**

Система призначена для підтримки процесу дослідження задач дробово-лінійного програмування в умовах невизначеності.

### **Цілі та задачі розробки**

Метою даної системи є спрощення процесу дослідження задачі дробово-лінійного програмування в умовах невизначеності за рахунок проведення експериментів та візуалізації результатів аналізу, що дозволить зменшити час, що витрачає дослідник на виявлення нових властивостей.

## **Висновок до розділу**

У даному розділі було описано предметне середовище програмного продукту,

визначені цілі та задачі розробки. Були проаналізовані наявні програмні аналоги та було визначено, що схожих по функціоналу програмних продуктів наразі не існує, тож були описані наявні аналоги, що вирішують окремі задачі, що і розроблювана система. Також були створені сценарії використання даного програмного продукту.

# Інформаційне забезпечення

## **Вхідні дані**

Для розв’язку індивідуальної задачі дробово-лінійного програмування в умовах невизначеності за допомогою системи користувачу необхідно ввести такі вхідні дані:

* кількість альтернативних значень коефіцієнтів ;
* набір коефіцієнтів ;
* набір коефіцієнтів ;
* набір експертних вагів ;
* набір максимально можливих значень дельта ;
* напрямок оптимізації;
* обмеження задачі;

Або наступні дані для випадкової генерації числових значень умови задачі:

* кількість змінних задачі;
* кількість обмежень задачі;
* кількість альтернативних значень коефіцієнтів ;
* тип неперервного розподілу для генерації числових значень задачі;
* числові значення параметрів для обраного неперервного розподілу;
* напрямок оптимізації;

Для виконання дослідження задачі дробово-лінійного програмування в умовах невизначеності за допомогою системи користувачу необхідно ввести такі вхідні дані:

* кількість експериментів;
* тип експерименту, що визначає яку саме залежність необхідно дослідити;
* кількість альтернативних значень коефіцієнтів ;
* набір коефіцієнтів ;
* набір коефіцієнтів ;
* набір експертних вагів ;
* набір максимально можливих значень відхилення оптимальних значень часткових цільових функцій від компромісного розв’язку ;
* напрямок оптимізації;
* обмеження задачі;
* числове значення кроку для змінюваного вхідного параметра;

Або наступні дані для виконання дослідження за допомогою випадкової генерації числових значень умови задачі:

* кількість експериментів;
* кількість змінних задачі;
* кількість обмежень задачі;
* кількість альтернативних значень коефіцієнтів ;
* тип неперервного розподілу для генерації числових значень задачі;
* числові значення параметрів для обраного неперервного розподілу;

## **Вихідні дані**

Вихідними даними як результат розв’язання індивідуальної задачі дробово-лінійного програмування в умовах невизначеності є:

1. Документ з розширенням .csv, що містить розв’язок задачі, а саме такі значення як:
   1. оптимальні значення змінних задачі дробово-лінійного програмування в умовах невизначеності ;
   2. оптимальні значення змінних;
   3. оптимальне значення допоміжної цільової функції ;
   4. оптимальні значення часткових цільових функцій ;
   5. значення , що показує наскільки компромісний розв’язок відрізняється від кожного оптимального значення часткових цільових функцій;
2. Документ з розширенням .csv, що містить умову задачі, а саме такі значення як:
   1. набір коефіцієнтів ;
   2. набір коефіцієнтів ;
   3. набір експертних вагів ;
   4. набір максимально можливих значень відхилення оптимальних значень часткових цільових функцій від компромісного розв’язку ;
   5. напрямок оптимізації;
   6. обмеження задачі;

Вихідними даними як результат дослідження задачі дробово-лінійного програмування в умовах невизначеності є:

1. Документ з розширенням .csv, що містить числові значення тих параметрів, що необхідно було дослідити:
2. Документ з розширенням .csv, що містить умову задачі, а саме такі значення як:
   1. набір коефіцієнтів ;
   2. набір коефіцієнтів ;
   3. набір експертних вагів ;
   4. набір максимально можливих значень дельта ;
   5. напрямок оптимізації;
   6. обмеження задачі;

## **Опис структури бази даних**

Детальна схема реляційної бази даних наведена у графічних матеріалах. На рисунку 2.1 зображені зв’язки між таблицями.

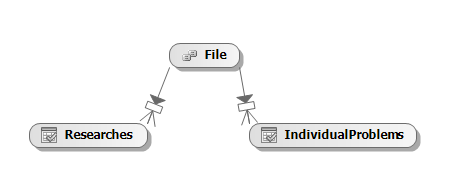


Рисунок 2.1 – схема зв’язків між таблицями

У таблицях 2.1 – 2.3 наведений детальний опис кожної таблиці бази даних.

Таблиця 2.1 – Опис таблиці “Індивідуальні задачі”

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Опис | Додаткові властивості |
| Id | INTEGER | Ідентифікатор таблиці | Первинний ключ, обов’язкове поле |
| Name | TEXT | Ім’я задачі | Обов’язкове поле |
| XCount | INTEGER | Кількість змінних | Обов’язкове поле |
| ConstraintsCount | INTEGER | Кількість обмежень | Обов’язкове поле |
| OptimizationDirection | INTEGER | Напрямок оптимізації | Обов’язкове поле |
| AlternativesCount | INTEGER | Кількість альтернатив | Обов’язкове поле |

Продовження Таблиці 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Опис | Додаткові властивості |
| Comment | TEXT | Коментар | - |
| FileInputIndividualProblemId | INTEGER | Ідентифікатор файлу, що містить умову задачі | Зовнішній ключ, обов’язкове поле |
| FileOutputIndividualProblemId | INTEGER | Ідентифікатор файлу, що містить розв’язок задачі | Зовнішній ключ, обов’язкове поле |

Таблиця 2.2 – Опис таблиці “Дослідження”

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Опис | Додаткові властивості |
| Id | INTEGER | Ідентифікатор таблиці | Первинний ключ, обов’язкове поле |
| Name | TEXT | Ім’я задачі | Обов’язкове поле |
| XCount | INTEGER | Кількість змінних | Обов’язкове поле |
| ConstraintsCount | INTEGER | Кількість обмежень | Обов’язкове поле |
| OptimizationDirection | INTEGER | Напрямок оптимізації | Обов’язкове поле |
| AlternativesCount | INTEGER | Кількість альтернатив | Обов’язкове поле |
| Comment | TEXT | Коментар | - |

Продовження Таблиці 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Опис | Додаткові властивості |
| ExperimentsCount | INTEGER | Кількість експериментів | Обов’язкове поле |
| Distribution | INTEGER | Вид неперервного рівномірного розподілу | Обов’язкове поле |
| Type | INTEGER | Тип експерименту | Обов’язкове поле |
| FileInputResearchId | INTEGER | Ідентифікатор файлу, що містить умову задачі | Зовнішній ключ, обов’язкове поле |
| FileOutputResearchId | INTEGER | Ідентифікатор файлу, що містить числові значення досліджуваних параметрів задачі | Зовнішній ключ, обов’язкове поле |

Таблиця 2.3 – Опис таблиці “Файли”

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Опис | Додаткові властивості |
| Id | INTEGER | Ідентифікатор таблиці | Первинний ключ, обов’язкове поле |
| Name | TEXT | Назва файлу | Обов’язкове поле |

## **Висновок до розділу**

У межах даного розділу були визначені усі необхідні вхідні і вихідні дані для розв’язку та дослідження в цілому задачі дробово-лінійного програмування в умовах невизначеності. Були описані атрибути та зв’язки між таблицями реляційної бази даних, визначені обов’язкові поля, первинні та зовнішні ключі.

# Математичне забезпечення

## **Змістовна постановка задачі**

Існують деякі методи розв’язання задачі дробово-лінійного програмування в умовах невизначеності. Необхідно реалізувати один із цих методі, створивши програмне забезпечення, що дозволить ефективно розв’язувати задачі такого типу, а також проводити дослідження і відкривати нові властивості.

## **Математична постановка задачі**

Задача комбінаторної оптимізації в умовах невизначеності має вигляд [1, 2]:



де — числа, — *i*-та довільна числовая характеристика допустимого розв’язку    — множина допустимих розв’язків. Під невизна-ченністю тут розуміється невизначенність значень коефіцієнтів 

В роботах [1, 2] були викладені основи конструктивної теорії знаходження компромісного розв’язку для такого класу задач.

Задача дробово-лінійного програмування (ЗДЛП) у детермінованій постановці має вигляд:



,

де , , ,   — дійсні числа; — змінні задачи.

Для того, щоб при поясненнях уникнути необхідності розгляду множини різних можливих варіантів, припустимо, що на  накладаються такі обмеження, при яких знаменник в (2) строго додатній для всіх допустимих значень  а також, що максимум є скінченним [3]:





Отже, існує R наборів коефіцієнтів   можливих значень коефіцієнтів  Тоді задача (2) перетворюється на задачу дробово-лінійного програмування в умовах невизначеності (ЗДЛПУН). За наявності невизначенності ставиться задача знаходження так званого компромісного розв’язку задачі [1, 2].

## **Обґрунтування методу розв’язання**

В умовах невизначності досить складно зрозуміти, який підхід до розв’язку задачі є накращим. Тож розглянемо декілька компромісних критеріїв оцінки розв’язку ЗДЛПУН.

**Критерий 1**

Знайти компромісний розв’язок  що задовольняє (3) і для якого виконується

 (4)

де , ,  - експертні вагові коефіцієнти.

**Критерій 2**

Введемо випадкову величину , де (n+1)-мірна дискретна випадково величина  задається таблицею



.

Знайти компромісний розв’язок  що задовольняє (3) і для якого виконується

 (5)

**Критерій 3**

Знайти компромісний розв’язок  що задовольняє (3) і для якого виконується



де для задачі на мінімум:



а для задачі на максимум відповідно:



**Критерій 4**

Якщо компромісного розв’язку, що задовольняє критерію A не існує, то знайти що задовольняє (3) на якому досягається



де  - відомі експертні вагові коефіцієнти.

**Критерій 5**

Для одного з наборів вагів  знайти оптимальний розв’язок задачі

 (9)

, (10)

якому відповідало б

 (11)

де ,  - відомі експертні вагові коефіцієнти.

## **Опис методів розв’язання**

Для обраної роботи були обрані Критерій 3 та Критерій 4, далі будемо їх називати Критерій А та Критерій Б відповідно. Далі розглянемо знаходження компромісного розв’язку за цими двома критеріями.

Компромісний розв’язок ( ) за критеріями А та Б (якщо за критерієм Б розв’язку не існує) знаходиться за розв’язком наступної ЗЛП:









Якщо вихідна задача (2) - (3) є задачею на максимум, то в задачі (14) - (17) нерівності (17) мають вигляд:



Зміст величин , :  - величина, яка показує наскільки ми повинні “посунутися” (поступитись) від встановленого допустимого відхилення від найкращого значення го часткового часткового функціоналу у випадку якщо не задовольняються обмеження (6).

Рисунок 1 ілюструє зв’язок основних елементів теоретичних положень знаходження компромісного розв’язку задачі в недетемінованій постановці.

Chart, radar chart

Description automatically generated

Рисунок 1 – Ілюстрація основних елементів теоретичних положень

## **Висновок до розділу**

В даному розділі була описана змістовна та математична постановка задачі. Було описано декілька компромісних критеріїв оцінки розв’язку ЗДЛПУН та обрані ті, що найкраще відповідають наявним умовам.

# Програмне та технічне забезпечення

## **Засоби розробки**

**.NET**

Для розробки програмного забезпечення для цієї роботи був обраний .NET.

.NET - це фреймворк для розробки програмного забезпечення та екосистема, розроблена та підтримувана корпорацією Майкрософт для спрощення проектування десктоп та веб-додатків. Це популярна безкоштовна платформа, яка в даний час широко використовується для багатьох типів застосунків, оскільки вона надає інструменти для більшості етапів розробки програмного забезпечення.

Основні мови програмування, що використовують .NET розробники це C#, F#, and Visual Basic.

**C#**

Мову для розробки даного застосунку було обрано C#.

C# (вимовляється як "сі шарп") - сучасна об'єктно-орієнтовна і типобезечна мова програмування. C# дозволяє розробникам створювати безліч типобезпечних і надійних програм, які є частиною екосистеми .NET. C# відноситься до широко відомого сімейства мов C, і буде добре знайомою кожному, хто працював раніше з Java, C, C ++, або JavaScript.

C# - це об'єктно- і компонентно- орієнтована мова програмування. Вона надає мовні конструкції для безпосередньої підтримки такої концепції роботи. Завдяки цьому C# підходить для створення і застосування програмних компонентів. З моменту створення ця мова збагатилася функціями для підтримки нових робочих навантажень і сучасними рекомендаціями по розробці програмного забезпечення.

Ось лише кілька функцій мови C #, які дозволяють створювати надійні та стійкі застосунки:

* Очистка сміття - автоматично звільняє пам'ять, зайняту недоступними невикористовуваними об'єктами;
* Типи, що допускають значення null, забезпечують захист від змінних, які не посилаються на виділені об'єкти;
* Обробка винятків надає структурований і розширюваний підхід до виявлення помилок і відновлення після них;
* Лямбда-вирази підтримують прийоми функціонального програмування;
* Синтаксис LINQ створює загальний шаблон для роботи з даними з будь-якого джерела;
* Підтримка мов для асинхронних операцій надає синтаксис для створення розподілених систем;

У C# діє єдина система типів - всі типи, включаючи примітивні, такі як int і double, успадковуються від одного кореневого типу object. Всі типи використовують загальний набір операцій, а значення будь-якого типу можна зберігати, передавати і обробляти схожим чином. Також C # підтримує універсальні методи, що забезпечують підвищену безпеку типів і загальну продуктивність.

**UWP**

UWP технологія, що є частиною інфраструктури .NET, була основою для створення даної системи.

UWP забезпечує загальну систему типів, API та шаблони для застосунків для всіх пристроїв, що працюють під управлінням Windows 10. UWP дозволяє розробляти універсальні програми для ПК, планшетів, Xbox, Surface Hub, HoloLens або Інтернету речей (IoT).

Основні переваги застосунків UWP:

* Безпека. Програми UWP повідомляють, до яких ресурсів пристрою та даних вони здійснюють доступ. Користувач повинен дозволити такий доступ;
* Можливість використовувати загальний API на всіх пристроях під управлінням Windows 10;
* Можливість використання можливостей окремих пристроїв та адаптації користувацького інтерфейсу до різних розмірів екранів, розширень та щільностей;
* Доступність в магазині Microsoft Store, на всіх пристроях, що працюють під управлінням Windows 10;
* Можливість встановити та видалити без ризику для комп'ютера;
* Можливість програмування на C #, C ++, Visual Basic і JavaScript. Для користувацького інтерфейсу можна використовувати WinUI, XAML, HTML або DirectX;

**XAML**

Для розробки користувацького інтерфейсу була використана мова XAML.

XAML ("Extensible Application Markup Language") - це мова розмітки, розроблена корпорацією Майкрософт і використовується для створення інтерфейсів додатків. Він схожий на HTML, який визначає вміст веб-сторінки.

Як і інші мови розмітки, XAML використовує теги для визначення об'єктів. Приклад цього можна бачити на Рисунку 4.1. Теги можуть бути вкладені в інші теги для визначення об'єктів усередині об'єктів. Атрибути об’єкта, такі як назва, розмір, форма та колір, визначаються в тегах.

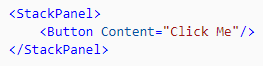


Рисунок 4.1 – Приклад коду на XAML

**SQLite**

При роботі з базою даних застосунку використовувалось SQLite.

SQLite - це бібліотека, що реалізує автономний, безсерверний, транзакційний механізм баз даних SQL. Код SQLite знаходиться у відкритому доступі і, отже, безкоштовний для будь-яких цілей, комерційних або приватних. SQLite - це найпоширеніша база даних у світі.

SQLite - це вбудований механізм баз даних SQL. На відміну від більшості інших баз даних SQL, SQLite не має окремого серверного процесу. SQLite читає та пише безпосередньо на звичайні файли диска. Повна база даних SQL з кількома таблицями, індексами, тригерами та поданнями міститься в одному дисковому файлі. Формат файлу бази даних є міжплатформеним - ви можете вільно копіювати базу даних між 32-розрядною та 64-розрядною системами або між архітектурами big-endian та little-endian.

## **Вимоги до технічного забезпечення**

Даний програмний продукт рекомендується використовувати на IBM-сумісному комп’ютері, що має характеристики не нижче ніж:

* Процесор Intel Core i5-7200U;
* Оперативна пам’ять – 4 Гб;
* Жорсткий диск – 200 Гб.

Програмні засоби, що необхідно використовувати при використанні програмного продукту:

* Операційна система Windows 10.

## **Архітектура програмного забезпечення**

Розроблюваний програмний продукт має трирівневу архітектуру.

Трирівнева архітектура - це клієнт-серверна архітектура, в якій функціональні процеси користувацького інтерфейсу, бізнес-логіки та рівня доступу до даних розробляються та підтримуються як незалежні модулі. Основними перевагами такої архітектури є:

* Масштабованість. Сервери програм можна розгорнути на багатьох машинах, база даних більше не вимагає підключення від кожного клієнта.
* Багаторазове використання. Ви можете повторно використовувати рівень бізнес-логіки (BLL), тобто середній рівень з різними інтерфейсами користувача, такими як веб-форма ASP.NET, веб-служба ASP.NET, віконна форма .NET тощо. Також є можливим повторно використовувати рівень доступу до бази даних (DAL).
* Поліпшення цілісності даних. BLL, тобто середній рівень, може гарантувати, що в базу даних дозволяється вставляти, оновлювати або видаляти лише дійсні дані.
* Покращення безпеки. BLL загалом безпечніший, оскільки розміщується на більш захищеному центральному сервері.
* Прихована структура бази даних. Оскільки фактична структура бази даних прихована від користувача.

Слід зазначити, що на верхньому рівні, що відповідає за взаємодію з користувачем використовується паттерн розробки MVVM. Як і багато інших шаблонів проектування, MVVM допомагає впорядковувати код і розбивати програми на модулі, щоб спростити і пришвидшити розробку, оновлення та повторне використання коду.

Ідея шаблону полягає в розділення коду на View, ViewModel і Model:

* View - це колекція видимих ​​елементів, яка також отримує введені користувачем дані. Сюди входять користувацькі інтерфейси (UI), анімація та текст. View не взаємодіє безпосередньо на елементи, що представлені на ньому щоб їх змінити
* ViewModel знаходиться між шарами View та Model. Тут розташовані елементи керування для взаємодії з View, тоді як прив’язка використовується для підключення елементів інтерфейсу користувача у View до елементів керування у ViewModel.
* Модель містить логіку програми, яка отримується за допомогою ViewModel після отримання вхідних даних від користувача через View.

В графічних матеріалах представлена структурна схема програмного забезпечення, де можна бачити описана вище архітектуру представлена у графічному вигляді.

### **Діаграма класів**

На рисунку 4.1 зображена діаграма класів проєкту PresentationLevel, що знаходиться на рівні представлення.

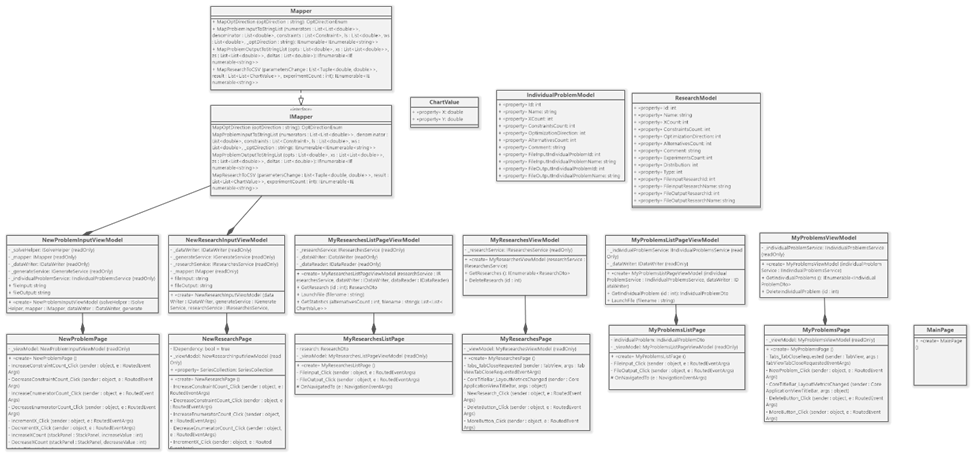


Рисунок 4.1

В таблиці 4.1 дамо опис типів, що зображені на Рисунку 4.1.

Таблиця 4.1

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| Mapper | Клас | Містить імплементацію інтерфейсу IMapper |
| IMapper | Інтерфейс | Містить методи для мапінгу моделей між рівнем бізнес логіки та рівнем представлення |
| ChartValue | Клас | Містить поля для представлення точки на графіку |
| IndividualProblemModel | Клас | Модель, що містить поля для опису індивідуальної задачі |
| ResearchModel | Клас | Модель, що містить поля для опису дослідження |
| ConstraintSymbolStruct | Структура | Містить поля для опису обмеження задачі дробово-лінійного програмування |
| OptimizationStruct | Структура | Містить поля для опису напряму опимізації |
| MyProblemsListPageViewModel | Клас | Містить поля та методи для зв’язку MyProblemsListPage з рівнем бізнес логіки |
| MyProblemsViewModel | Клас | Містить поля та методи для зв’язку MyProblemsPage з рівнем бізнес логіки |
| MyResearchesListPageViewModel | Клас | Містить поля та методи для зв’язку MyResearchesListPage з рівнем бізнес логіки |
| MyResearchesViewModel | Клас | Містить поля та методи для зв’язку MyResearchesPage з рівнем бізнес логіки |
| NewProblemInputViewModel | Клас | Містить поля та методи для зв’язку NewProblemPage з рівнем бізнес логіки |
| NewResearchInputViewModel | Клас | Містить поля та методи для зв’язку NewResearchPage з рівнем бізнес логіки |
| MyProblemsListPage | Клас | Сторінка, що містить інформацію про конкретну збережену індивідуальну задачу |
| MyProblemsPage | Клас | Сторінка, що містить список із усіх збережених індивідуальних проблем |
| NewProblemPage | Клас | Сторінка, що містить поля для можливості введеня умови індивідуальної задачі та її розв’язання |
| MyResearchesListPage | Клас | Сторінка, що містить інформацію про конкретне збережене дослідження |
| MyResearchesPage | Клас | Сторінка, що містить список з усіх збережених досліджень |
| NewResearchPage | Клас | Сторінка, що містить поля для можливості введеня конфігурації для проведення дослідження |
| GeneralNavigationPage | Клас | Містить логіку для переключення між сторінками |
| MainPage | Клас | Головна сторінка, що містить останні збережені індивідуальні задачі та дослідження |

На рисунку 4.2 зображена діаграма класів проєкту CSVManager, що знаходиться на рівні бізнес-логіки.

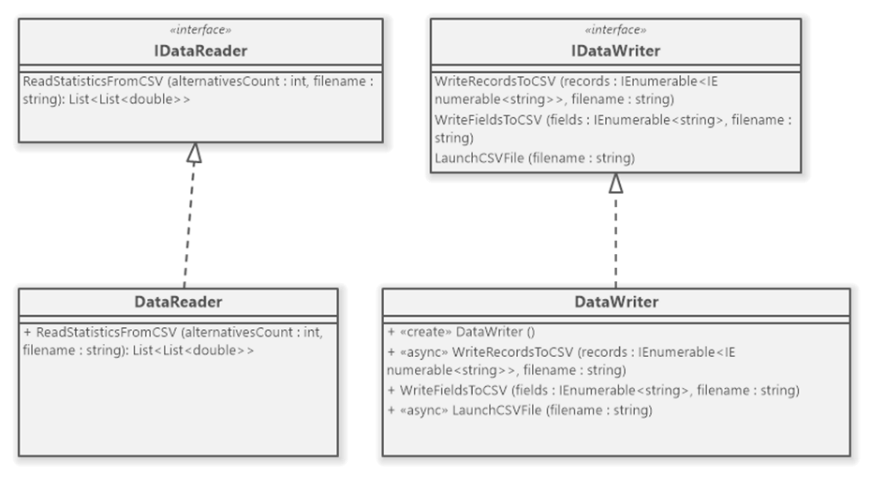


Рисунок 4.2

В таблиці 4.2 дамо опис типів, що зображені на Рисунку 4.2.

Таблиця 4.2

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| IDataReader | Інтерфейс | Містить методи для читання файлів формату .csv |
| DataReader | Клас | Містить методи, що реалізують інтерфейс IDataReader |
| IDataWriter | Інтерфейс | Містить методи для запису даних до файлу формату .csv |
| DataWriter | Клас | Містить методи, що реалізують інтерфейс IDataWriter |

На рисунку 4.3 зображена діаграма класів проєкту ProblemGenerator, що знаходиться на рівні бізнес-логіки.

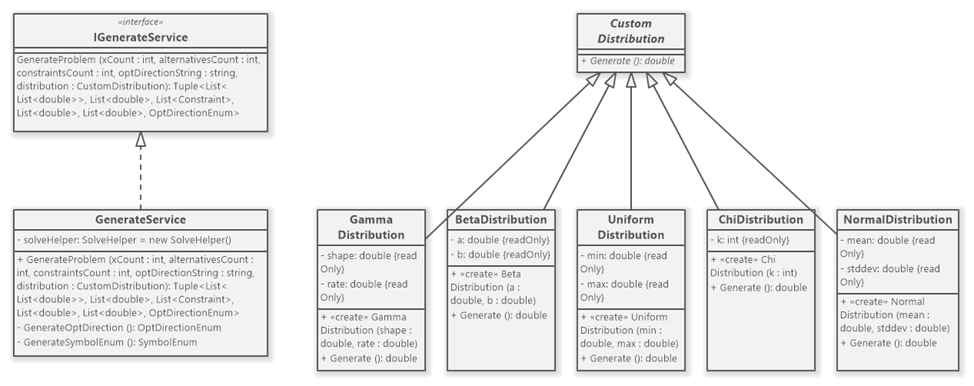


Рисунок 4.3

В таблиці 4.3 дамо опис типів, що зображені на Рисунку 4.3.

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| IGenerateService | Інтерфейс | Містить методи для випадкової генерації вхідних даних індивідуальної задачі |
| GenerateService | Клас | Сервіс, що містить методи, що імплементують інтерфейс IGenerateService |
| CustomDistribution | Клас | Модель, що описує абстрактний неперервний розподіл |
| BetaDistribution | Клас | Модель, що описує бета-розподіл |
| ChiDistribution | Клас | Модель, що описує розподіл хі-квадрат |
| GammaDistribution | Клас | Модель, що описує гамма-розподіл |
| NormalDistribution | Клас | Модель, що описує нормальний розподіл |
| UniformDistribution | Клас | Модель, що описує рівномірний розподіл |

На рисунку 4.4 зображена діаграма класів проєкту ProblemManager, що знаходиться на рівні бізнес-логіки.

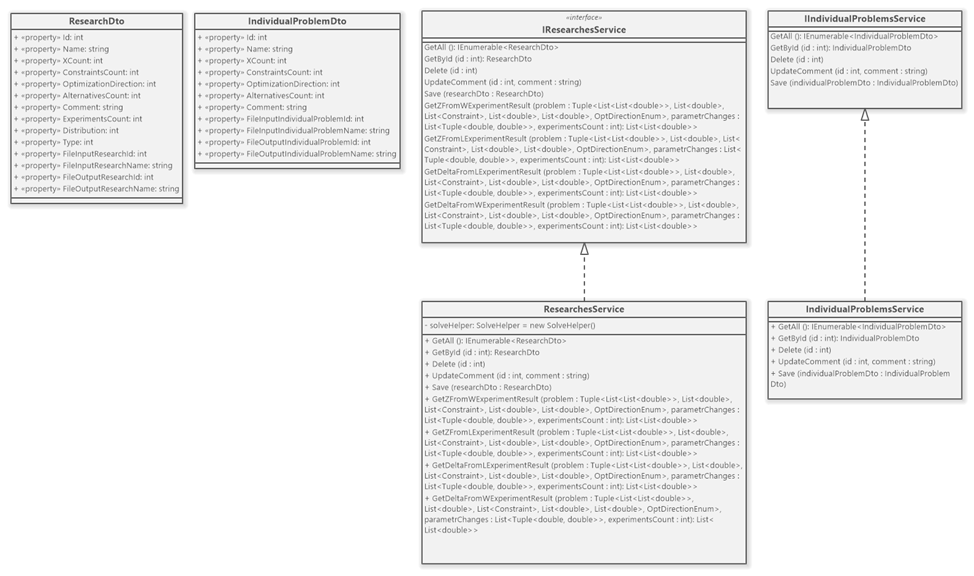


Рисунок 4.4

В таблиці 4.4 дамо опис типів, що зображені на Рисунку 4.4.

Таблиця 4.4

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| IndividualProblemDto | Клас | Модель, що містить поля для опису індивідуальної задачі |
| ResearchDto | Клас | Модель, що містить поля для опису сутності дослідження |
| IIndividualProblemsService | Інтерфейс | Містить методи, що описують CRUD операції для індивідуальних задач |
| IResearchesService | Інтерфейс | Містить методи, що описують CRUD операції для досліджень |
| IndividualProblemsService | Клас | Містить методи, що імплементують інтерфейс IIndividualProblemsService |
| ResearchesService | Клас | Містить методи, що імплементують інтерфейс IResearchesService |

На рисунку 4.5 зображена діаграма класів проєкту ProblemSolver, що знаходиться на рівні бізнес-логіки.

Text

Description automatically generated with medium confidence

Рисунок 4.5

В таблиці 4.5 дамо опис типів, що зображені на Рисунку 4.5.

Таблиця 4.5

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| ISolveHelper | Інтерфейс | Містить методи для розв’язку задачі дробово-лінійного програмування в умовах невизначеності |
| SolveHelper | Клас | Містить методи, що імплементують інтерфейс ISolveHelper |
| SymbolEnum | Перерахування | Містить перерахування символів для обмеження задачі |
| OptDirectionEnum | Перерахування | Містить перерахування для напрямків оптимізації задачі |
| Constraint | Клас | Модель, що містить поля для опису сутності обмеження |

На рисунку 4.6 зображена діаграма класів проєкту DataAccess, що знаходиться на рівні доступу до даних.

A picture containing text

Description automatically generated

Рисунок 4.6

В таблиці 4.6 дамо опис типів, що зображені на Рисунку 4.5.

Таблиця 4.6

|  |  |  |
| --- | --- | --- |
| Назва | Тип | Опис |
| Research | Клас | Модель, що містить поля для опису сутності дослідження |
| IndividualProblem | Клас | Модель, що містить поля для опису сутності індивідуальної задачі |
| File | Клас | Модель, що містить поля для опису сутності файл |

### **Діаграма послідовності**

### **Діаграма компонентів**

Діаграма компонентів розроблюваного продукту наведена у графічних додатках. Там ми можемо бачити, що програма поділена на три логічних рівня: рівень представлення, бізнес-логіки та доступу до даних. Рівень доступу до даних містить моделі, що представляють таблиці бази даних та репозиторії для взаємодії з базою даних. Рівень бізнес-логіки містить сервіси, що в свою чергу реалізують логіку програми, що прописана у вимогах. Рівень представлення зроблений за патерном MVVM, тому містить такі компоненти як Model, View, ViewModel.

### **Специфікація функцій**

У таблиці 4.7 наведено опис основних функцій розроблюваного програмного продукту.

Таблиця 4.7

|  |  |  |
| --- | --- | --- |
| Клас | Функція | Опис |
| DataReader | List<List<double>> ReadStatisticsFromCSV (int alternativesCount, string filename) | Читає дані для подальшого відображення на графіках з файлу формату .csv |
| DataWriter | void WriteRecordsToCSV(IEnumerable<IEnumerable<string>> records, string filename) | Записує дані у файл формату .csv |
| GenerateService | Tuple<List<List<double>>, List<double>, List<Constraint>, List<double>, List<double>, OptDirectionEnum> GenerateProblem(int xCount, int alternativesCount, int constraintsCount, string optDirectionString, CustomDistribution distribution) | Випадково генерує і повертає вхідні дані для ЗДЛПУН |
| IndividualProblemsService | IEnumerable<IndividualProblemDto> GetAll() | Повертає усі збережені індивідуальні задачі |
| IndividualProblemDto GetById(int id) | Повертає індивідуальну задачу по індексу |
| void Delete(int id) | Видаляє індивідуальну задачу по індексу |
| void Save(IndividualProblemDto individualProblemDto) | Зберігає індивідуальну задачу |
| ResearchesService | IEnumerable<ResearchDto> GetAll() | Повертає усі збережені дослідження |
| ResearchDto GetById(int id) | Повертає дослідження по індексу |
| void Delete(int id) | Видаляє дослідження по індексу |
| void Save(ResearchDto researchDto) | Зберігає дослідження |
| List<List<double>> GetZFromWExperimentResult(Tuple<List<List<double>>, List<double>, List<Constraint>, List<double>, List<double>, OptDirectionEnum> problem, List<Tuple<double, double>> parametrChanges, int experimentsCount) | Повертає списки значень, що описують как змінюються величини при зміні величин |
| List<List<double>> GetZFromLExperimentResult(Tuple<List<List<double>>, List<double>, List<Constraint>, List<double>, List<double>, OptDirectionEnum> problem, List<Tuple<double, double>> parametrChanges, int experimentsCount) | Повертає списки значень, що описують как змінюються величини при зміні величин |
| List<List<double>> GetDeltaFromLExperimentResult(Tuple<List<List<double>>, List<double>, List<Constraint>, List<double>, List<double>, OptDirectionEnum> problem, List<Tuple<double, double>> parametrChanges, int experimentsCount) | Повертає списки значень, що описують как змінюються величини при зміні величин |
| List<List<double>> GetDeltaFromWExperimentResult(Tuple<List<List<double>>, List<double>, List<Constraint>, List<double>, List<double>, OptDirectionEnum> problem, List<Tuple<double, double>> parametrChanges, int experimentsCount) | Повертає списки значень, що описують как змінюються величини при зміні величин |
| SolveHelper | SolveProblem(List<List<double>>numerator, List<double>denominator,List<Constraint>\_constraints, List<double> ls, List<double> ws, OptDirectionEnum \_optDirection) | Функція, що розв’язує задачу дробово-лінійного програмування в умовах невизначенності. |
| FindDeltas(List<double>xs,List<List<double>>numerators, List<double> denominator, List<double> fOpts) | Функція, що обчислює дельти. |
| Solve(List<Constraint>\_constraints,List<double>\_function, OptDirectionEnum \_optDirection, int xCount, int zCount) | Функція, що що розв’язує задачу дробово-лінійного програмування. |

## **Опис звітів**

В даній системі є можливість будувати графіки чотирьох видів, кожний з яких відповідає окремому типу дослідження, а саме:

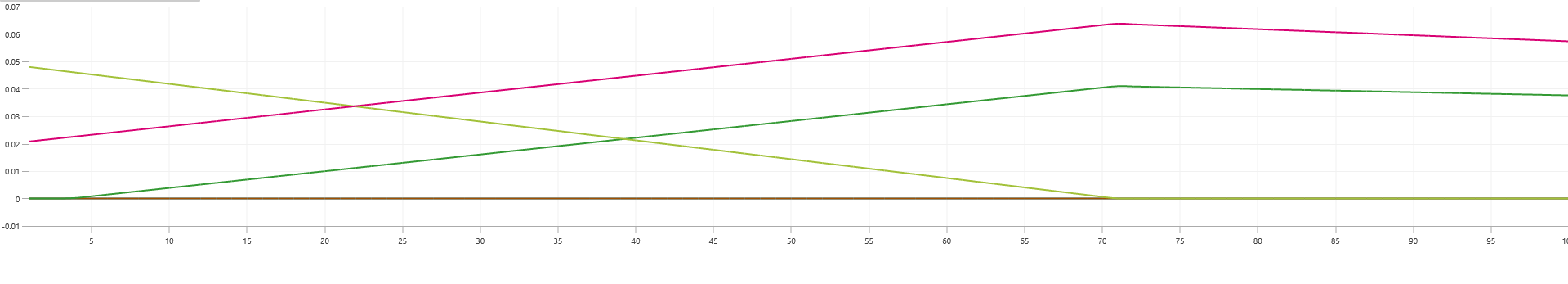
* Дослідження залежності z від l
* Дослідження залежності z від w
* Дослідження залежності delta від l
* Дослідження залежності delta від w

Для того, щоб проілюструвати описані вище графіки залежностей, проведемо декілька експериментів та наведемо їх конфігурацію в Таблиці 4.8.

Таблиця 4.8

|  |  |
| --- | --- |
| Назва | Значення |
| Кількість експериментів | 100 |
| Кількість змінних в кожній ЗДЛПУН | 5 |
| Кількість альтернативних наборів коефіцієнтів чисельника ЗДЛПУН | 5 |
| Кількість обмежень ЗДЛПУН | 5 |
| Вид неперервного розподілу | Рівномірний (мінімальне значення – 1, максимальне - 1000) |

На рисунку 4.7 та 4.8 представлені графіки залежностей z від l для напрямку оптимізації на максимум та мінімум відповідно.



## **Висновок до розділу**

*Дати висновок до цього розділу.*

# Технологічний розділ

## **Керівництво користувача**

Після завантаження програми користувач потрапляє на головну сторінку, що зображена на Рисунку 5.1. На цій є можливість завантажити файл з основними означеннями та теоретичними положеннями, що зустрічаються у програмі. Фрагмент цього файлу наведений на Рисунку 5.2.

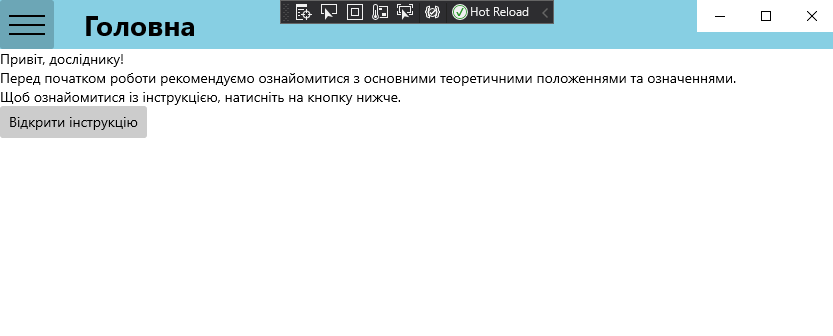


Рисунок 5.1

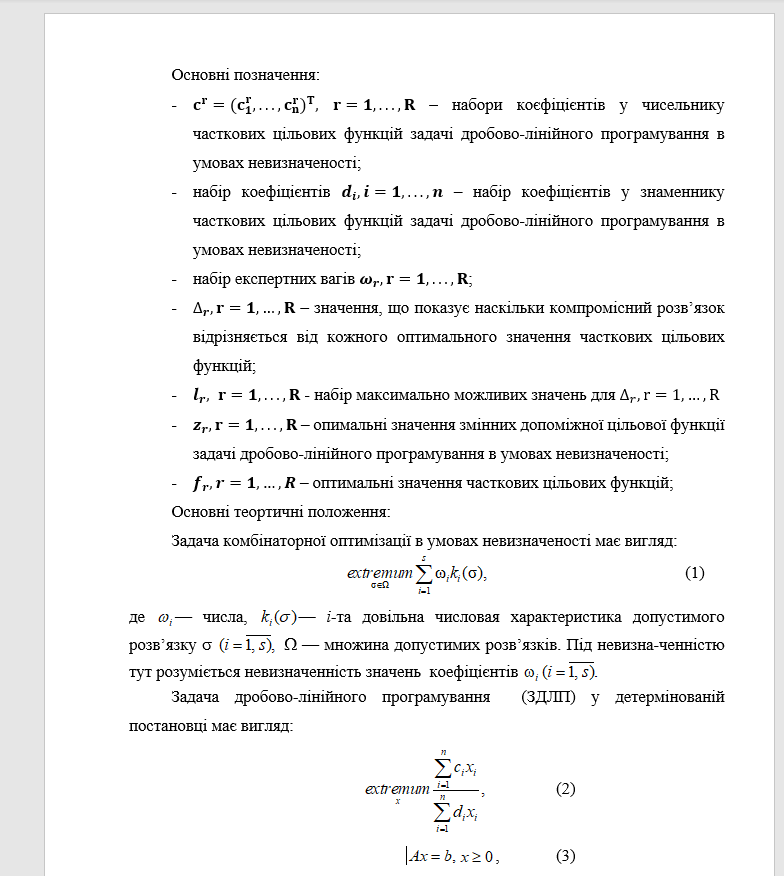


Рисунок 5.2

На рисунку 5.3 зображено навігаційне меню за допомогою якого можна перейти на сторінку, де відображений список з усіх збережених індивідуальних задач (Рисунок 5.4) або на сторінку, де відображений список з усіма збереженими дослідженнями (Рисунок 5.5).

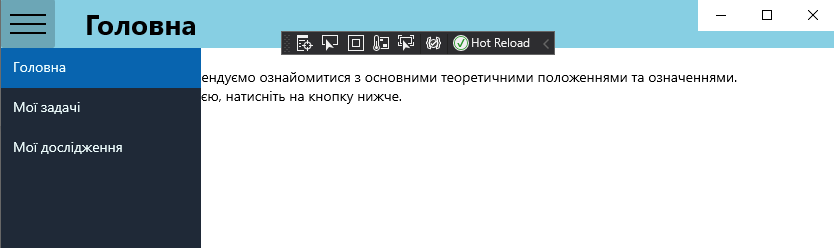


Рисунок 5.3



Рисунок 5.4



Рисунок 5.5

Зі сторінки на Рисунку 5.4 є можливість створити нову індивідуальну задачу натиснувши кнопку “Нова задача”. В такому разі користувач потрапить на сторінку створення нової ЗДЛПУН, що зображена на Рисунку 5.6.

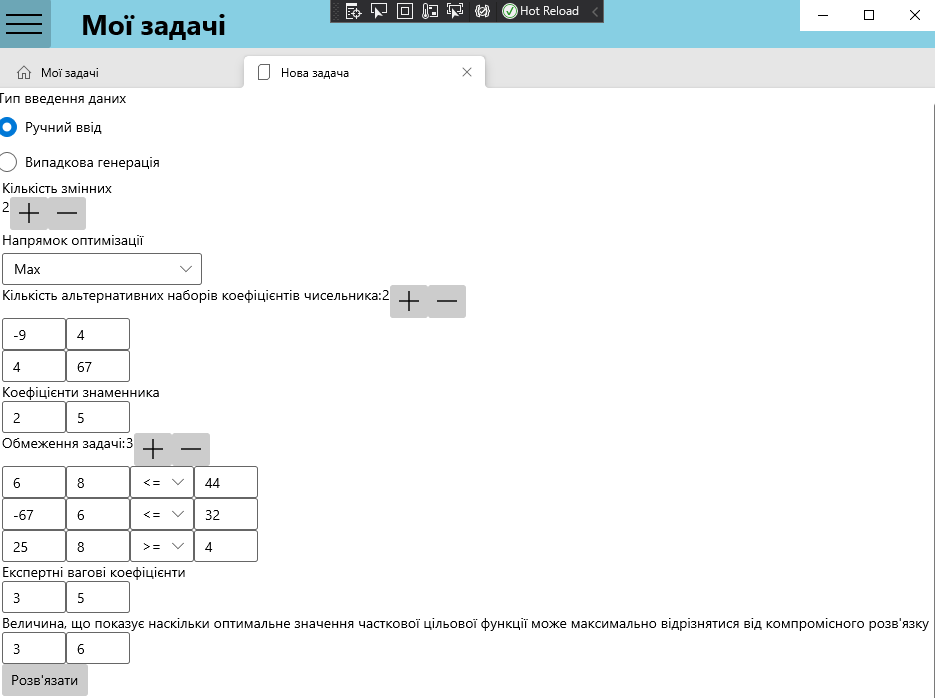


Рисунок 5.6

Окрім ручного введення умови є можливість генерації вхідних даних за допомогою одного з п’яти видів неперервного розподілу (рівномірний, нормальний, хі-квадрат, бета-розподіл, гамма-розподіл). Для цього потрібно змінити тип введення даних на “Випадкова генерація” (Рисунок 5.7).

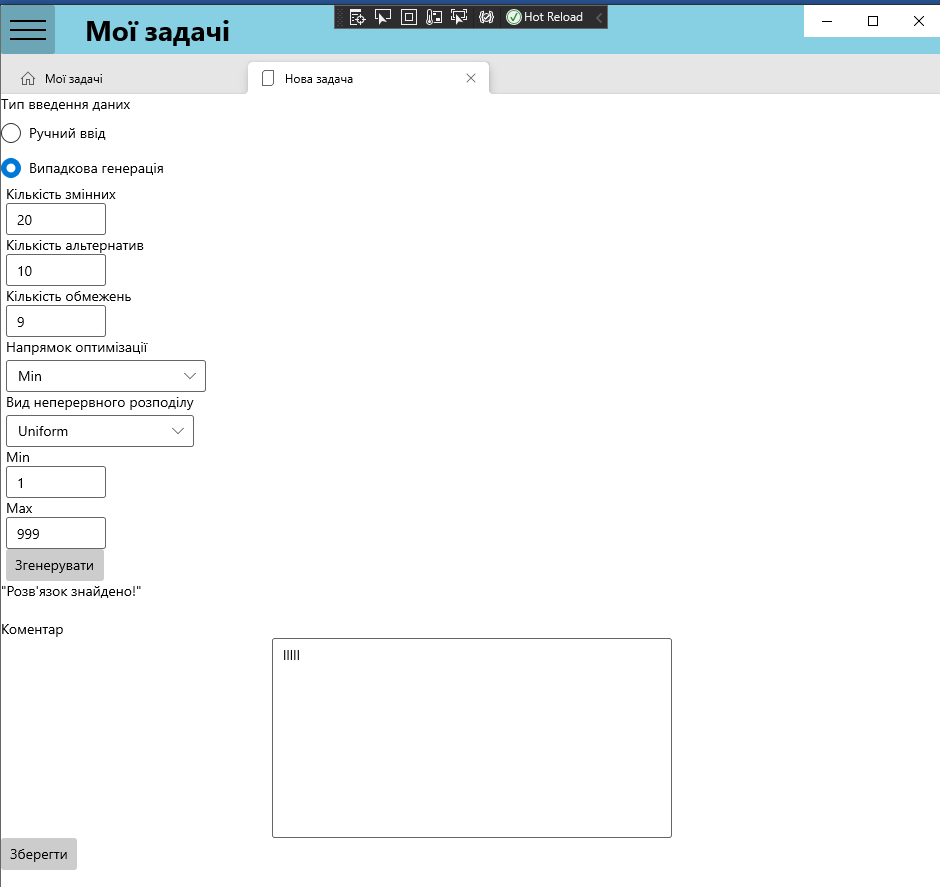
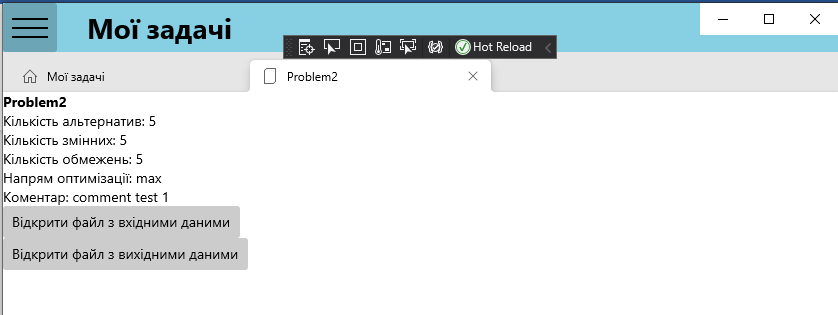


Рисунок 5.7

На Рисунку 5.8 зображена сторінка детального перегляду збереженої індивідуальної задачі. За допомогою кнопок “Відкрити файл з вхідними даними” та “Відкрити файл із вихідними даними” можна продивитися вхідні дані (Рисунок 5.9) та вихідні дані (Рисунок 5.10) ЗДЛПУН у файлах з розширенням .csv.



Приклад 5.8

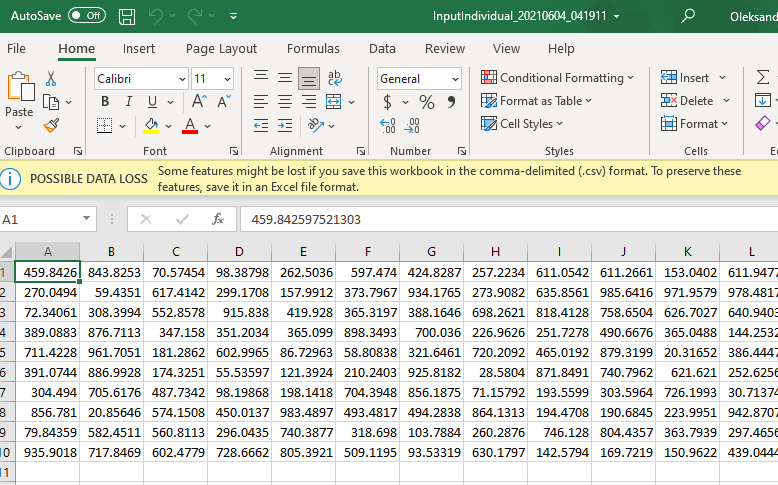


Рисунок 5.9

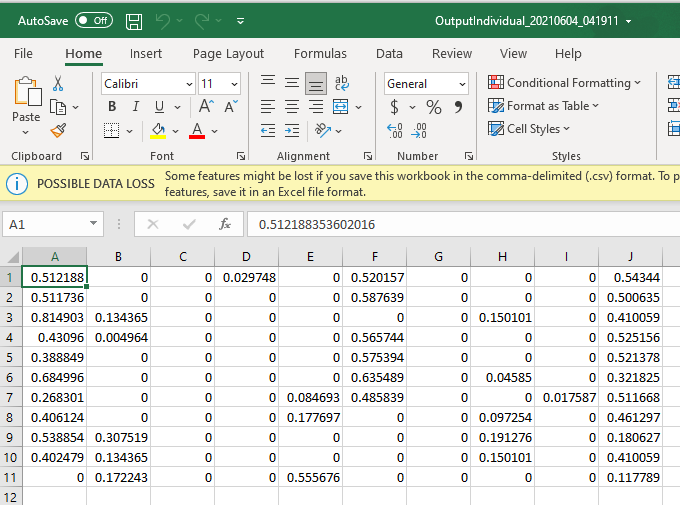


Рисунок 5.10

На Рисунку 5.11 зображена сторінка для створення нового дослідження. Ми можемо ввести вхідні дані для ЗДЛПУН як самостійно (Рисунок 5.11), так і згенерувавши необхідні дані (Рисуок 5.12) таким самим способом як і у випадку сторінки для стовореня нової індивідуальної задачі.

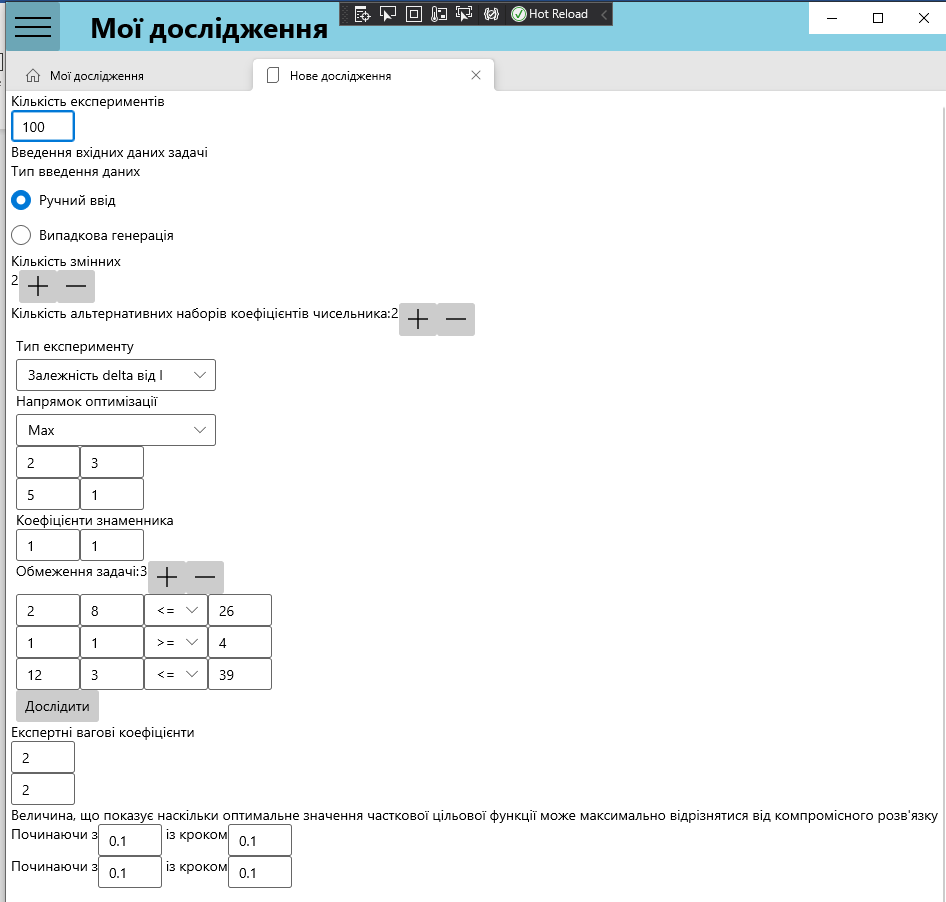


Рисунок 5.11

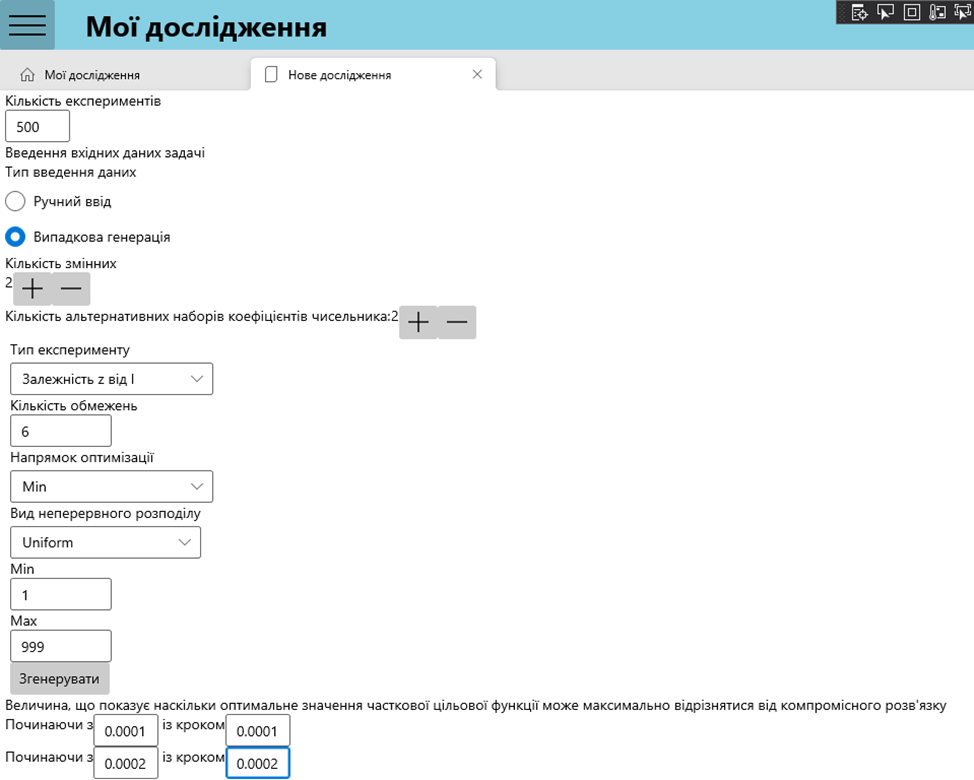


Рисунок 5.12

На Рисунку 5.13 зображена сторінка детального перегляду збереженого дослідження.

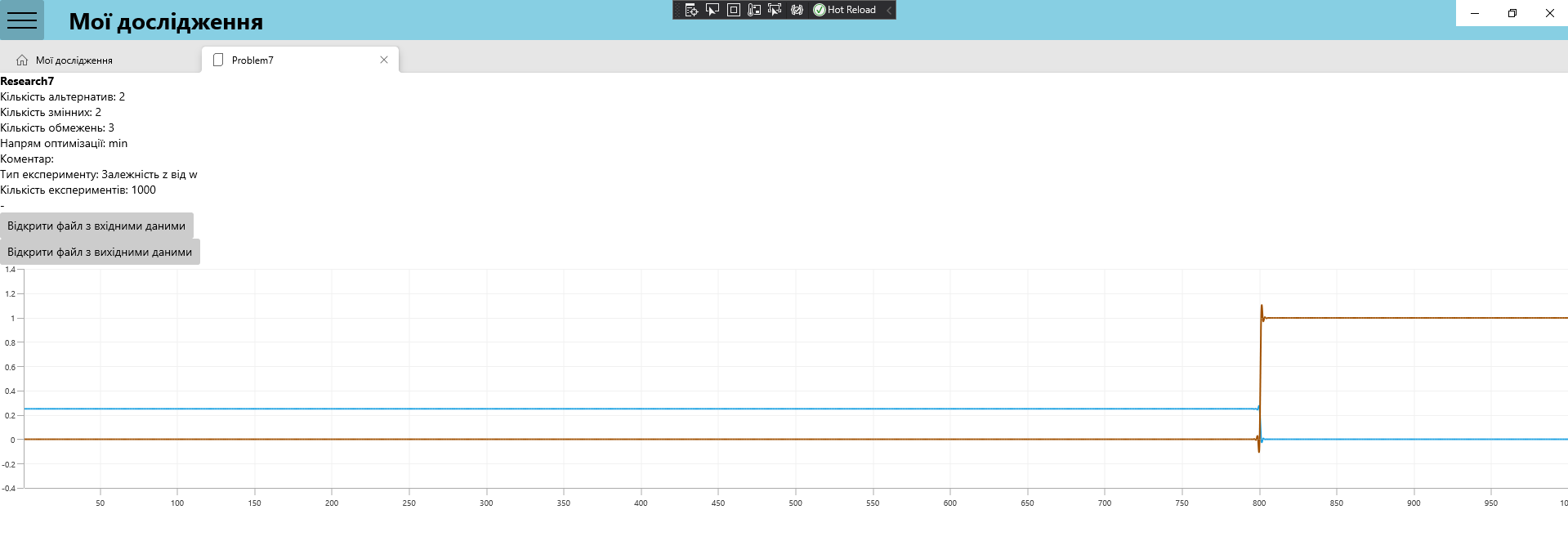


Рисунок 5.13

За допомогою кнопок “Відкрити файл з вхідними даними” та “Відкрити файл із вихідними даними” можна продивитися вхідні дані (Рисунок 5.14) та вихідні дані (Рисунок 5.15) ЗДЛПУН у файлах з розширенням .csv.

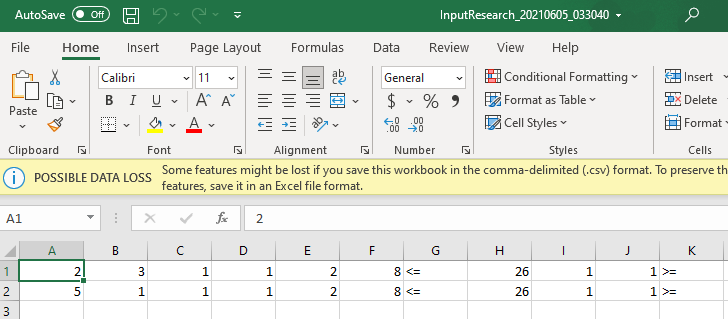


Рисунок 5.14

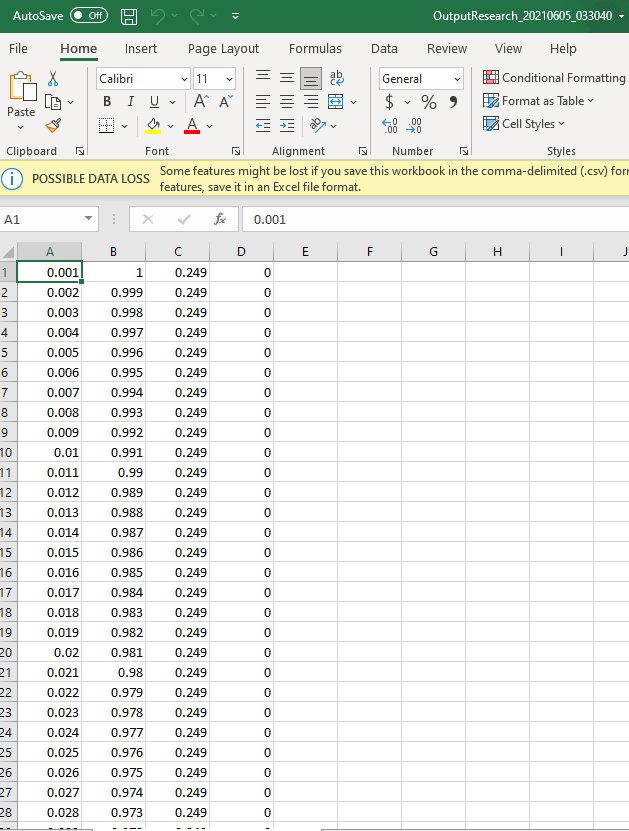


Рисунок 5.15

## **Випробування програмного продукту**

### **Мета випробувань**

Метою випробувань є перевірка того, що функціонал інформаційної системи з підтримки процесу дослідження задачі дробово-лінійного програмування в умовах невизначеності відповідає вимогам технічного завдання.

### **Загальні положення**

Випробування проводяться на основі наступних документів:

* ГОСТ 34.603−92. Інформаційна технологія. Види випробувань автоматизованих систем;
* ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### **Результати випробувань**

Для перевірки коректної роботи, було проведено тестування основних функцій, що описані в технічному завданні.

Таблиця 5.1

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу розв’язання ЗДЛПУН попередньо ввівши вхідні дані вручну |
| Початковий стан | Відкрита сторінка “Нова задача” для розв’язання індивудуальної задачі |
| Дія | Обрати тип введення даних “Ручний ввід”; ввести необхідну кількість змінних, напрямок оптимізації, набори коефіцієнтів чисельника, коефіцієнти знаменника, обмеження задачі, експертні вагові коефіцієнти, величини, що показують наскільки оптимальні значення часткової цільової функції можуть відрізнятися від компромісного розв’язку; натиснути кнопку “Розв’язати” |
| Очікуваний результат | З’явилося повідомлення “Розв’язок знайдено!” у випадку коли задача має розв’язок або “Задача не має розв’язку” у випадку коли задача не має розв’язку; у випадку коли розв’язок знайдено: відкрилися файли у фоматі .csv із вхідними та вихідними даними задачі, з’явилося поле для введення коментара та кнопка “Зберегти” для збереження індивідуальної задачі |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.2

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу розв’язання ЗДЛПУН попередньо згенерувавши вхідні дані |
| Початковий стан | Відкрита сторінка “Нова задача” для розв’язання індивудуальної задачі |
| Дія | Обрати тип введення даних “Випадкова генерація”; ввести необхідну кількість змінних, напрямок оптимізації, кількість альтернатив, кількість обмежень, вид неперервного розподілу та його параметри; натиснути кнопку “Згенерувати” |
| Очікуваний результат | З’явилося повідомлення “Розв’язок знайдено!” у випадку коли задача має розв’язок або “Задача не має розв’язку” у випадку коли задача не має розв’язку; у випадку коли розв’язок знайдено: відкрилися файли у фоматі .csv із вхідними та вихідними даними задачі, з’явилося поле для введення коментара та кнопка “Зберегти” для збереження індивідуальної задачі |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.4

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу перегляду вхідних даних збереженої ЗДЛПУН у файлі формату .csv |
| Початковий стан | Відкрита сторінка з інформацією про конкретну збережену індивідуальну задачу |
| Дія | Натиснути на кнопку “Відкрити файл із вхідними даними” |
| Очікуваний результат | Відкрився файл з розширенням .csv де відображені вхідні дані ЗДЛПУН |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.5

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу перегляду вихідних даних збереженої ЗДЛПУН у файлі формату .csv |
| Початковий стан | Відкрита сторінка з інформацією про конкретну збережену індивідуальну задачу |
| Дія | Натиснути на кнопку “Відкрити файл із вихідними даними” |
| Очікуваний результат | Відкрився файл з розширенням .csv де відображені вихідні дані ЗДЛПУН |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.6

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу проведення експериментів над ЗДЛПУН попередньо ввівши вхідні дані вручну |
| Початковий стан | Відкрита сторінка “Нове дослідження” |
| Дія | Обрати тип введення даних “Ручний ввід”; ввести дані в усі необхідні поля; натиснути кнопку “Дослідити” |
| Очікуваний результат | З’явилося повідомлення “Дослідження проведено!” у випадку коли задача має розв’язок або “Задача не має розв’язку” у випадку коли задача не має розв’язку; у випадку коли розв’язок знайдено: відкрилися файли у фоматі .csv із вхідними та вихідними даними екперименту, з’явилося поле для введення коментара та кнопка “Зберегти” для збереження дослідження |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.7

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу проведення експериментів над ЗДЛПУН попередньо згенерувавши вхідні дані |
| Початковий стан | Відкрита сторінка “Нове дослідження” |
| Дія | Обрати тип введення даних “Випадкова генерація”; ввести дані в усі необхідні поля; натиснути кнопку “Дослідити” |
| Очікуваний результат | З’явилося повідомлення “Дослідження проведено!” у випадку коли задача має розв’язок або “Задача не має розв’язку” у випадку коли задача не має розв’язку; у випадку коли розв’язок знайдено: відкрилися файли у фоматі .csv із вхідними та вихідними даними екперименту, з’явилося поле для введення коментара та кнопка “Зберегти” для збереження дослідження |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.9

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу перегляду вхідних даних ЗДЛПУН для дослідження у файлі формату .csv |
| Початковий стан | Відкрита сторінка з інформацією про конкретне збережене дослідження |
| Дія | Натиснути на кнопку “Відкрити файл із вхідними даними” |
| Очікуваний результат | Відкрився файл з розширенням .csv де відображені вхідні дані ЗДЛПУН для дослідження |
| Чи співпадає очікуваний результат з фактичним | Так |

Таблиця 5.10

|  |  |
| --- | --- |
| Мета | Перевірка функціоналу перегляду вихідних даних ЗДЛПУН для дослідження у файлі формату .csv |
| Початковий стан | Відкрита сторінка з інформацією про конкретне збережене дослідження |
| Дія | Натиснути на кнопку “Відкрити файл із вихідними даними” |
| Очікуваний результат | Відкрився файл з розширенням .csv де відображені вихідні дані ЗДЛПУН для дослідження |
| Чи співпадає очікуваний результат з фактичним | Так |

## **Висновок до розділу**

*Дати висновок до цього розділу.*

ЗАГАЛЬНІ ВИСНОВКИ

Тут потрібно вказати усі висновки з усіх розділів ПЗ – але не механічним перенесенням тексту висновків, а узагальнивши усі основні результати, отримані в ДП та надавши рекомендації із застосування отриманих результатів та їх практичної цінності

Перелік посилань

*При написанні переліку посилань використовуйте відповідні вимоги.*

Додаток А

*ДП 6101.00.000 ПЗ*

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

16

ДП 7101.00.000 ПЗ

(Найменування програми (документа))

***Тексти програмного коду***

***Система моніторингу стану працездатності банкоматної мережі***

(Вид носія даних)

*DVD-R*

(Обсяг програми (документа) , арк., Кб)

*73 арк, 244 Кб*

Київ – 2021 року