МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Спеціальність 126 Інформаційні системи та технології

ЗВІТ

з переддипломної практики бакалаврів
на тему:

„ Інформаційна система з підтримки процесу дослідження задачі дробово-лінійного програмування в умовах невизначеності ”

Місце проходження практики: ТОВ “ЕПАМ СИСТЕМЗ”

Виконав студент ІС-71 Вознюк Олександра Віталіївна ____________
(шифр групи, прізвище, ім’я, по батькові) (підпис)

Керівник практики
від університету Тєлишева Тамара Олексіївна ____________
(прізвище, ім’я, по батькові) (підпис)

Керівник практики
від підприємства Бабейко Надія Анатоліївна ____________
(прізвище, ім’я, по батькові) (підпис)

Дата захисту ____________________ Оцінка „__________________”

Київ 2021

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 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]:

$$F = \sum_{j=1}^{n} c_j x_j,$$

де c_j - задані постійні величини.

При умовах:

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \ \ (i = \overline{1,k}),$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \ \ (i = \overline{k+1,m}),$$

$$x_j \ge 0 \ \ (j = \overline{1,l},\ l \le n),$$

де a_{ij}, b_i, c_j - задані постійні величини і $k \le m$.

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:

$$F = \frac{\sum_{i=1}^{n} c_i^r x_i}{\sum_{i=1}^{n} d_i x_i}, \ r = \overline{1,R},$$

де c_i, d_i - задані постійні величини.

Під невизначеністю в даному випадку розуміється існування R можливих значень c_i, $i = \overline{1,n}$.

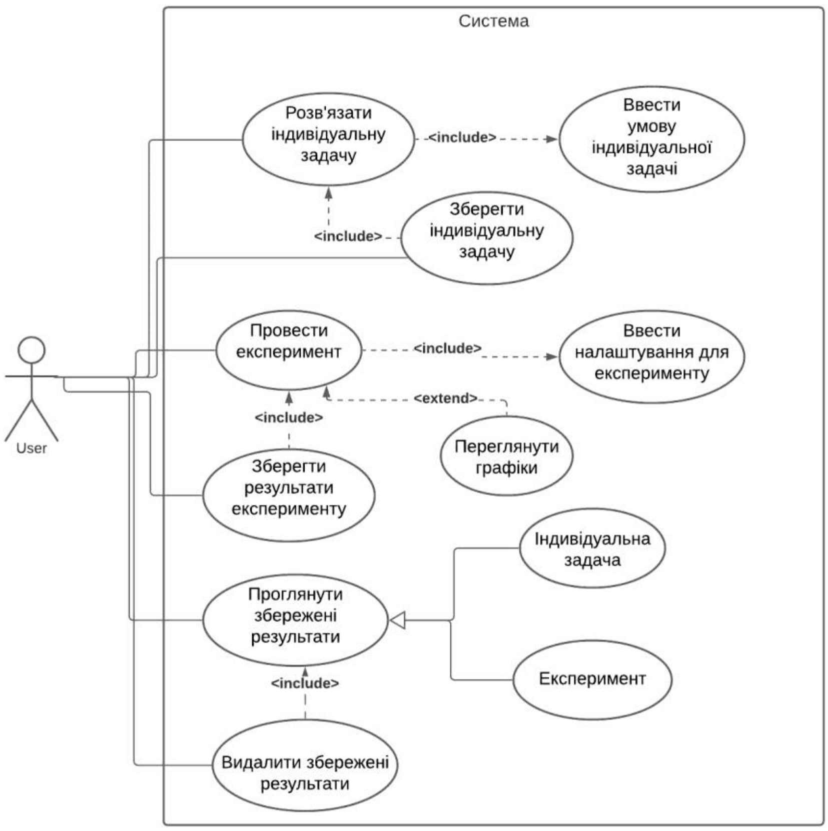
1.1.1 Опис процесу діяльності

Продукт, що розробляється, є десктоп застосунком призначений для спрощення процесу дослідження задач дробово-лінійного програмування в умовах невизначеності.

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.

1.1.2 Опис функціональної моделі

Нижче наведена UML-діаграма варіантів використання, де описано як актор (дослідник) буде взаємодіяти із системою.



1.2 Огляд наявних аналогів

Так як не виявлено програмного забезпечення, що має інструменти для вирішення та дослідження задачі дробово-лінійного програмування в умовах невизначеності в межах даної теми доцільно аналізувати програмні засоби, що можуть бути корисні при вирішенні ЗЛП, тому що ця дія повторюється не один раз впродовж моделювання даної задачі.

1.2.1 Google OR-Tools

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#.

1.2.2 LINGO Software

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.

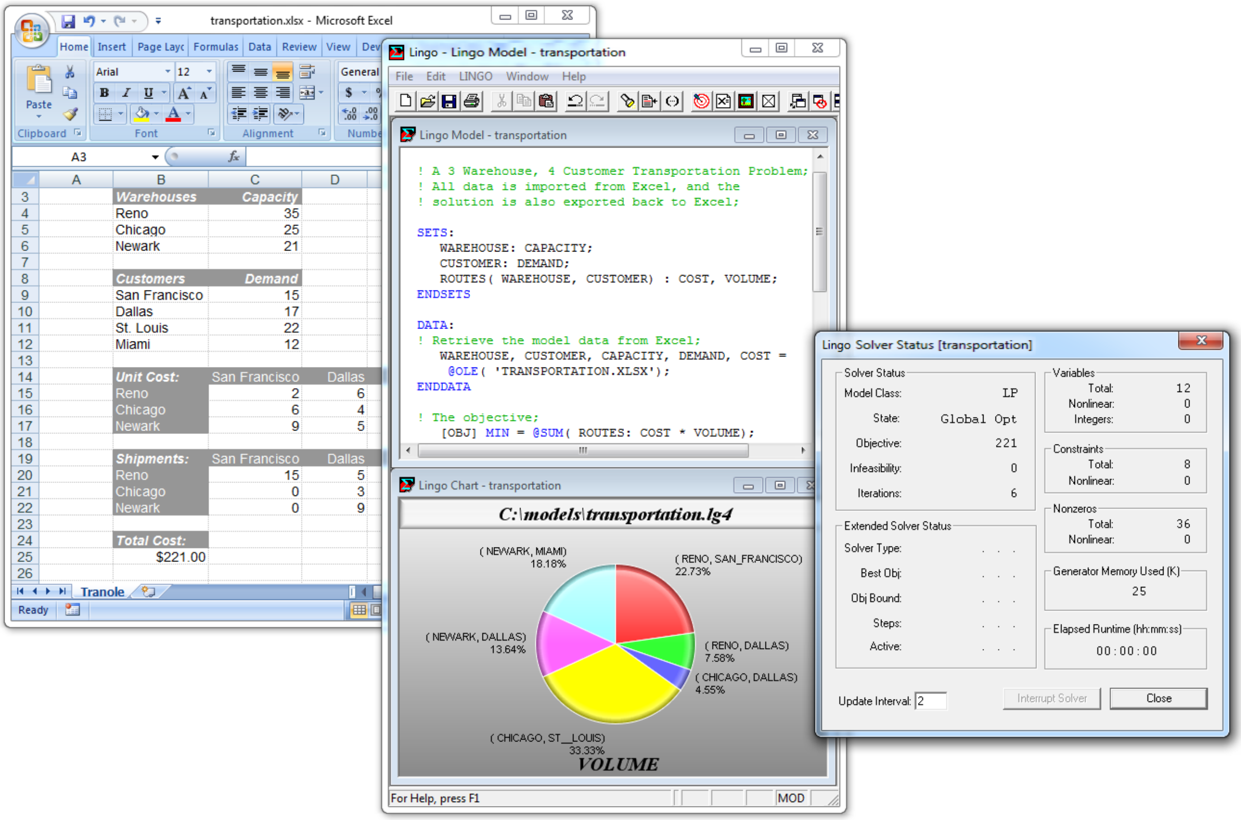


Рисунок 1 – Приклад інтерфейсу LINGO Software

1.3 Постановка задачі

1.3.1 Призначення розробки

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

1.3.2 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, описані та проаналізовані розроблені програмні продукти, що мають схожий функціонал. Також були сформовані цілі та задачі розробки і створені сценарії використання даного програмного продукту.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

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

- Набір коефіцієнтів $\mathrm{c}^{\mathrm{r}} = (\mathrm{c}_1^{\mathrm{r}}, \dots, \mathrm{c}_{\mathrm{n}}^{\mathrm{r}})^{\mathrm{T}}, \mathrm{r} = 1, \dots, \mathrm{R}$.
- Набір коефіцієнтів $d_i, i = 1, \dots, n$.
- Набір експертних вагів $\omega_r, \mathrm{r} = 1, \dots, \mathrm{R}$.
- Набір максимально можливих значень дельта $l_r,\ \mathrm{r} = 1, \dots, \mathrm{R}$.

Або наступні дані для випадкової генерації умови задачі:

- Кількість змінних задачі
- Кількість обмежень задачі
- R (кількість альтернативних значень для коефіцієнта коефіцієнтів $\mathrm{c}^{\mathrm{r}} = (\mathrm{c}_1^{\mathrm{r}}, \dots, \mathrm{c}_{\mathrm{n}}^{\mathrm{r}})^{\mathrm{T}}, \mathrm{r} = 1, \dots, \mathrm{R}$)
- Вид розподілу для генерації випадкових величин та необхідні параметри

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

- Тип експерименту (яку саме залежність необхідно дослідити)
- Кількість експериментів
- Кількість змінних задачі
- Кількість обмежень задачі
- R (кількість альтернативних значень для коефіцієнта коефіцієнтів $\mathrm{c}^{\mathrm{r}} = (\mathrm{c}_1^{\mathrm{r}}, \dots, \mathrm{c}_{\mathrm{n}}^{\mathrm{r}})^{\mathrm{T}}, \mathrm{r} = 1, \dots, \mathrm{R}$)
- Вид розподілу для генерації випадкових величин та необхідні параметри
- Крок з яким необхідно змінювати параметри задачі

2.2 Вихідні дані

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

оптимальні значення змінних задачі дробово-лінійного програмування в умовах невизначеності $x_i, \; i = 1, \ldots, n$.

z_r, r $= 1, \ldots,$ R,.

Оптимальне значення допоміжної цільової функції Z.

Δ_r, r $= 1, \ldots,$ R.

Значення часткових цільових функцій $f_r, r = 1, \ldots, R$.

Вихідними даними як результат дослідження задачі дробово-лінійного програмування для кожного експерименту є ті ж самі, що описані вище і додатково графік залежностей досліджуваних величин.

2.3 Опис структури бази даних

Наведемо перелік таблиць та їх полів у таблиці 2.1.

Таблиця 2.1 – Перелік таблиць та полів

№	Назва таблиці	Поля
1	Задачі	Id; Ім'я; Кількість змінних; Кількість обмежень; Напрямок оптимізації; Кількість альтернатив; Коментар;
2	Дослідження	Id; Ім'я; Кількість змінних; Кількість обмежень; Напрямок оптимізації; Кількість альтернатив;

		Коментар; Кількість експериментів; Тип експерименту; Вид розподілу;

2.4 Структура масивів інформації

Частина даних, що використовуються для зберігання вхідних та вихідних даних задачі дробово-лінійного програмування в умовах невизначеності зберігаються також у файлах з розширенням .csv. У файлах такого типу кожен рядок відповідає одному рядку таблиці, а записи розділяються спеціальними символами (зазвичай комами). У системі існує два типи таких файлів, які будуть зберігати:

- Вхідні дані задачі дробово-лінійного програмування в умовах невизначеності описані у (2.1)
- Вихідні дані задачі дробово-лінійного програмування в умовах невизначеності описані у (2.2)

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 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.

3.2 Математична постановка задачі

Недостатність інформації про умови та обмеження, що властиві багатьом системам приводить до необхідності розробляти нові методи прийняття рішень. В роботах [1, 2] були викладені основи конструктивної теорії знаходження компромісного рішення для класу задач комбінаторної оптимізації в умовах невизначеності виду

$$\underset{\sigma\in\Omega}{extremum} \sum_{i=1}^{s} \omega_i k_i(\sigma), \qquad (3.1)$$

де ω_i — числа, $k_i(\sigma)$ — i-та довільна числовая характеристика допустимого рішення σ $(i = \overline{1,\ s})$, Ω — множина допустимих рішень.

Під невизначенністю тут розуміється невизначенність значень коефіцієнтів $\omega_i (i = \overline{1,\ s})$.

В [3] підтверджена ефективність цих теоретичних положень на прикладі однопродуктової та багатопродуктової транспортних задач. Результати отримані в [1–3] поширюються також і на задачу дробно-лінійного програмування, де величини ω_i и $k_i(\sigma)$ $(i = \overline{1,\ s})$ зв’язні нелінійно і формально не належать до класу задач комбінаторної оптимізації.

Задача дробово-лінійного програмування у детермінованій постановці має вигляд:

$$\underset{x}{extremum} \frac{\sum_{i=1}^{n} c_i x_i}{\sum_{i=1}^{n} d_i x_i}, \qquad (3.2)$$

$$Ax = b, \qquad (3.3)$$

де, $c = (c_1, \dots, c_n)^T$, $d = (d_1, \dots, d_n)^T$, $b = (b_1, \dots, b_m)^T$, $A = (a_{ij})$, $i = 1, \dots, m$, $i = 1, \dots, n$ — дійсні числа, $x = (x_1, \dots, x_n)^T$, $x \geq 0$ - змінні задачи.

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

$\sum_{i=1}^{n} d_i x_i \neq 0$, (3.4)

$\sum_{i=1}^{n} d_i x_i > 0$, (3.5)

Отже, існує R наборів коефіцієнтів $\mathrm{c^r = (c_1^r, \dots, c_n^r)^T}$, $\mathrm{r = 1, \dots, R}$ можливих значень коефіцієнтів $\mathrm{c_i}$, $\mathrm{i = 1, \dots, n}$. Знайти за заданими компромісними критеріями рішення задачі (1)–(2) в умовах сформульованої вище невизначеності. Ціллю є знайти такий компромісний розв'язок, який би задовільняв усі альтернативи не менше ніж на якусь встановлену величину.

3.3 Опис методу розв'язання

Оскільки в умовах невизначеності достатньо складно визначити чіткі критерії ефективності рішень, у статті [4] пропонується декілька компромісних критеріїв оцінки рішення задачі, що розглядається. Для даної роботи були обрані критерії 3 та 4 описані в [4]. Будемо називати їх критерій А та В відповідно.

3.3.1 Критерій А

Знайти компромісне рішення $x = (x_1, \dots, x_n)^T$, що задовольняє (3), у якого:

$$\Delta_i \leq l_i,\ l_i \geq 0,\ i = 1, \dots, R, \tag{3.5}$$

де для задачі на мінімум:

$$\Delta_r = \frac{\sum_{i=1}^{n} c_i^r x_i}{\sum_{i=1}^{n} d_i x_i} - f_{opt}^r, r = 1, \dots, R \tag{3.6}$$

А для задачі на максимум:

$$\Delta_r = f_{opt}^r - \frac{\sum_{i=1}^{n} c_i^r x_i}{\sum_{i=1}^{n} d_i x_i}, r = 1, \dots, R \tag{3.7}$$

3.3.2 Критерій B

Якщо компромісного рішення, що задовольняє критерію A не існує, то знайти $x = (x_1, \dots, x_n)^T$, що задовольняє (3) на якому досягається

$$\min_x \sum_{r=1}^{R} \omega_r \, max\{0; \Delta_r - l_r\}, \tag{3.8}$$

де $\omega_r > 0$, $r = 1, \dots, R$ - відомі експертні вагові коефіцієнти.

3.3.3 Побудова компромісного рішення

Як відомо [5], задача (2)-(3) зводиться до задачі лінійного програмування (ЗЛП) наступним чином.

Введемо нові змінні

$$y_0 = \frac{1}{\sum_{i=1}^{n} d_i x_i}, \tag{3.9}$$

$$y_i = y_0 x_i, i = 1, \dots, n \tag{3.10}$$

Тоді задача (2)-(3) прийме вигляд

$$\underset{y,k}{extremum} \sum_{i=1}^{n} c_i \, y_i, \tag{3.11}$$

$$Ay = y_0 b, \tag{3.12}$$

$$\sum_{i=1}^{n} d_i \, y_i = 1, y \geq 0, \tag{3.13}$$

де $y = (y_1, \dots, y_n)^T$.

По рішенню ЗЛП (3.11)-(3.13) знаходиться оптимальне рішення задачі (2)-(3): $x_i = \frac{y_i}{y_0}, i = 1, \dots, n$. При цьому оптимальне значення функціоналів (2) і (1.11) приймають однакове значення.

3.3.4 Знаходження компромісного рішення за критеріями A та B

Компромісне рішення ($x_i = \frac{y_i}{y_0}$, $i = 1, \dots, n$) за критеріями A та B (якщо за критерієм A рішення не існує) знаходиться за рішенням наступної ЗЛП:

$$Z = \min_{y,z,k} \sum_{r=1}^{R} \omega_r \, z_r, \tag{3.14}$$

$$Ay = y_0 b, \tag{3.15}$$

$$\sum_{i=1}^{n} d_i\, y_i = 1, \tag{3.16}$$

$$\sum_{i=1}^{n} c_i^r\, y_i - z_r \le f_{opt}^r + l_r, r = 1, \dots, R \tag{3.17}$$

Якщо вихідна задача (2)-(3) є задачею на максимум, то в задачі (3.14)-(3.17) нерівності (3.17) мають вигляд:

$$\sum_{i=1}^{n} c_i^r\, y_i + z_r \ge f_{opt}^r - l_r, r = 1, \dots, R \tag{3.18}$$

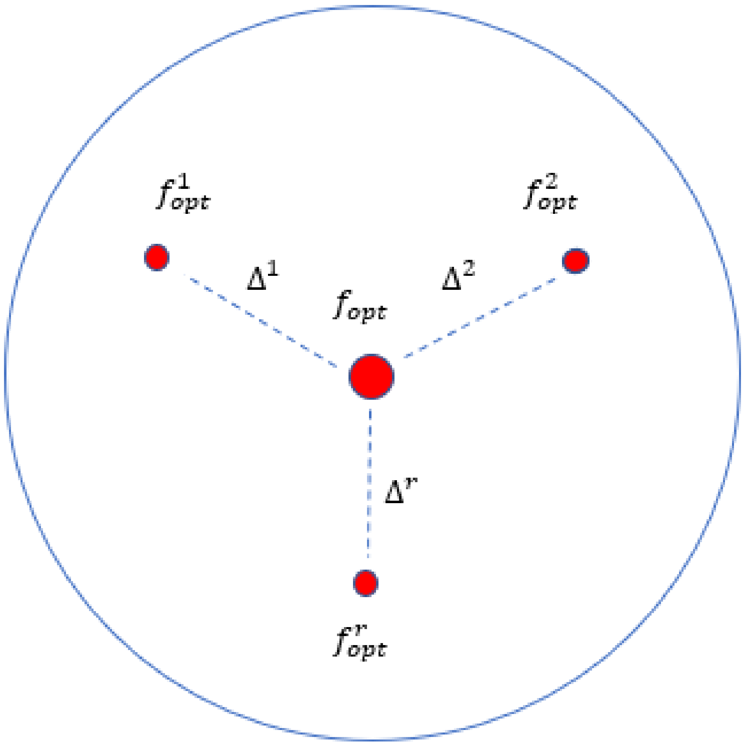


Рисунок 3.1 – Ілюстрація знаходження оптимального розв'язку при умові невизначеності

3.3.5 Приклади розв'язання задачі

3.3.5.1 Приклад розв'язання задачі у детермінованій постановці

Умова:

$$f = \frac{2x_1 + 3x_2}{x_1 + x_2} \to min$$

$$\begin{cases} 2x_1 + 8x_2 \le 26, \\ x_1 + x_2 \ge 4, \\ 12x_1 + 3x_2 \le 39. \end{cases}$$

$$x_1, x_2 \ge 0.$$

Розв'язання:

Введемо нові змінні

$y_0 = \frac{1}{x_1+x_2}$, $y_i = y_0 x_i$, $i = 1, \ldots, n$.

Тоді задача прийме вигляд

$$f = 2y_1 + 3y_2 \to min$$

$$\begin{cases} 2y_1 + 8y_2 - 26y_0 \le 0, \\ y_1 + y_2 - 4y_0 \ge 0, \\ 12y_1 + 3y_2 - 39y_o \le 0 \\ y_1 + y_2 = 1. \end{cases}$$

$$y_1, y_2 \ge 0$$

Таким чином ми отримали задачу лінійного програмування. Далі її можна вирішити симплекс-методом і тоді отримаємо $y_1 = 0.75$, $y_2 = 0.25$, $y_0 = 0.25$. Знайдемо кінцевий розв'язок за допомогою формули $x_i = \frac{y_i}{y_0}$, $i = 1, \ldots, n$:

$$x_1 = \frac{0.75}{0.25} = 3,$$

$$x_2 = \frac{0.25}{0.25} = 1,$$

$$f_{opt} = \frac{2 * 3 + 3 * 1}{3 + 1} = 2.25.$$

3.3.5.2 Приклад розв'язання задачі у недетермінованій постановці

Маємо два набори коефіцієнтів c_i, $i = 1,2$, тоді умова задачі прийме такий вигляд:

Таблиця 1.1 – Приклад розв'язання задачі

Умова	$f_1 = \frac{2x_1 + 3x_2}{x_1 + x_2} \to min$	$f_2 = \frac{5x_1 + 1x_2}{x_1 + x_2} \to min$

	$\begin{cases} 2x_1 + 8x_2 \leq 26, \\ x_1 + x_2 \geq 4, \\ 12x_1 + 3x_2 \leq 39. \end{cases}$ $x_1, x_2 \geq 0.$ $l_1 = 0.25$ $w_1 = 2$	$\begin{cases} 2x_1 + 8x_2 \leq 26, \\ x_1 + x_2 \geq 4, \\ 12x_1 + 3x_2 \leq 39. \end{cases}$ $x_1, x_2 \geq 0.$ $l_2 = 0.25$ $w_2 = 2$
Відповідь	$x_1 = 3,$ $x_2 = 1,$ $f_{opt}^1 = 2.25.$	$x_1 = 1,$ $x_2 = 3,$ $f_{opt}^2 = 2.$

Введемо нові змінні

$y_0 = \frac{1}{x_1+x_2}, y_i = y_0 x_i, i = 1, \dots, n.$

Тоді задача прийме вигляд

$$Z = 2z_1 + 2z_2 \to min$$

$$\begin{cases} 2y_1 + 8y_2 - 26y_0 \leq 0, \\ y_1 + y_2 - 4y_0 \geq 0, \\ 12y_1 + 3y_2 - 39y_o \leq 0 \\ y_1 + y_2 = 1. \\ 2y_1 + 3y_2 - z_1 \leq 2.25 + 0.25, \\ 5y_1 + 1y_2 - z_2 \leq 2 + 0.25 \end{cases}$$

$$y_1, y_2 \geq 0$$

Таким чином ми отримали задачу лінійного програмування. Далі її можна вирішити симплекс-методом і тоді отримаємо $y_1 = 0.3125$, $y_2 = 0.6875$, $y_0 = 0.25$.

Знайдемо кінцевий розв’язок за допомогою формули $x_i = \frac{y_i}{y_0}, i = 1, \dots, n$:

$$x_1 = \frac{0.3125}{0.25} = 1.25,$$

$$x_2 = \frac{0.6875}{0.25} = 2.75,$$

$$z_1 = 0.1875,$$

$$z_2 = 0,$$

$$Z = 0.375,$$

$$\Delta_1 = 0.4375,$$

$$\Delta_2 = 0.25.$$

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При створені системи використовувалися наступні технології:

- .NET Framework
- Microsoft SQL Server
- UWP

4.1.1 .NET Framework

Програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проєкті Mono (в рамках угоди між Microsoft з Novell), DotGNU[en].

Захист авторських прав відноситься до створення середовищ виконання (CLR — Common Language Runtime) для програм .NET. Компілятори для .NET випускаються багатьма фірмами для різних мов вільно.

.NET поділяється на дві основні частини — середовище виконання (по суті віртуальна машина) та інструментарій розробки.

Середовища розробки .NET-програм: Visual Studio .NET (C++, C#, J#), SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки .NET-програм. Застосовні програми також можна розроблювати в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-коду дозволяє отримати крос-платформність на рівні скомпільованого проєкту (в термінах .NET: збірка), а не на рівні сирцевого тексту, як, наприклад, в С. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

Слід зазначити, що один з перших JIT-компіляторів для Java був також розроблений фірмою Microsoft (тепер починаючи з версії J2SE 1.3 створеної 8 травня 2000 в Java використовується як основна, досконаліша багаторівнева компіляція — Sun HotSpot). Сучасна технологія динамічної компіляції дозволяє досягнути аналогічного рівня швидкодії з традиційними «статичними» компіляторами (наприклад, C++) і питання швидкодії часто залежить від якості того чи іншого компілятора.

4.1.2 Microsoft SQL Server

Система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі.

Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

4.1.3 UWP

Платформа, створена Microsoft і вперше представлена в Windows 10. Метою даної платформи є допомога у створенні універсальних програм, що запускаються як на Windows 10, Windows 10 Mobile і Windows 10 IoT без зміни в коді. Є підтримка створення таких додатків на C ++, C #, VB.NET і XAML. API реалізований в C ++ і підтримується в C ++, VB.NET, C #, F # і JavaScript. Розроблена як розширення для Windows Runtime (платформи, представленої в Windows Server 2012 і Windows 8), дозволяє запускати додатки на різних апаратних платформах.

4.2 Вимоги до технічного забезпечення

Даний програмний продукт рекомендується використовувати на IBM-сумісному комп'ютері, що включає:

- процесор Intel Core i5-7200U;
- оперативна пам'ять – 8 Гб;
- жорсткий диск – 200 Гб.

Програмні засоби, що рекомендується використовувати при використанні програмного продукту:

– операційна система Windows версії 10.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Нижче в таблиці 4.1 наводиться опис усіх класів розробленої бібліотеки для розв'язання задачі дробово-лінійного програмування в умовах невизначеності.

Таблиця 4.1 – Опис класів

Назва класу	Опис
SolveHelper	Клас, що містить методи для розв'язання задачі дробно-лінійного програмування в умовах невизначенності
RandProblemGenerator	Клас, що містить методи для генерування задач дробно-лінійного програмування на основі задланих меж значень параметрів та коефіцієнтів
Constraint	Клас, що описує сутність "Обмеження"
NoOptimumException	Клас, що описує помилку у випадку коли задача не має розв'язку
Y0IsNullException	Клас, що описує помилку у випадку коли y0 = 0
OptDirectionEnum	Перечислення, що представляє набір значень, для опису напрямку оптимізації
SymbolEnum	Перечислення, що представляє набір значень, для опису символів ">=", "=" та "<="
MyEnumExtensions	Клас, що дозволяя розширити функціонал класу SymbolEnum.

4.4 Специфікація функцій

Нижче в таблиці 4.2 наводиться опис основних функцій розробленої бібліотеки для розв'язання задачі дробово-лінійного програмування в умовах невизначеності.

Таблиця 4.2 – Опис основних функцій

Функція	Опис
SolveProblem(List<List<double>>numerators, List<double>denominator,List<Constraint>_constraints, List<double> ls, List<double> ws, OptDirectionEnum _optDirection)	Функція, що розв’язує задачу дробно-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 дробно-лінійного програмування.
GenerateProblem(List<Tuple<double,double>>coefBounds, List<Tuple<double, double>> bBounds, int constraintCount, string optDirectionString, List<Tuple<double, double>> lsBounds, List<Tuple<double, double>> wsBounds)	Функція, що генерує задачу дробно-лінійного програмування на основі введених обмежень на значення зміни вхідних даних.

Висновок до розділу

Були визначені та описані основні технології для розробки програмного забезпечення. Була створена структура реляційної базі даних та описані основні класи системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Pavlov A.A. Optimization for one class of combinatorial problems under uncertainty. *Адаптивні системи автоматичного управління*. 2019. **1**. № 34. С. 81–89. doi: 10.20535/1560-8956.1.2019.178233.
2. Pavlov A.A. Combinatorial optimization under uncertainty and formal models of expert estimation. *Вісник Національного технічного університету «ХПІ».* 2019. № 1. С. 3–7. doi: 10.20998/2079-0023.2019.01.01.
3. Павлов А.А., Жданова Е.Г. Транспортная задача в условиях неопределенности // Проблемы управления и информатики. – 2020. – № 2 – С.34-45.
4. Павлов А.А., Жданова Е.Г. Задача дробно-линейного программирования в условиях неопределенности.
5. Г. Вагнер. Основы исследования операций, том 2. С.381.
6. И. Л. Акулич. Математическое программирование в примерах и задачах. 1986. С. 11-12.