НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №4
із дисципліни «Системний аналіз»
на тему
«Методи і процедури прийняття рішень у багатокритеріальних задачах»

Виконав:
студент групи КМ-02
Пилипченко Б. О.

Перевірила:
доцент кафедри ПМА
Вовк Л. Б.

Київ - 2024

Зміст

Вступ ... 3
Мета ... 3
Порядок виконання роботи ... 3
Завдання на лабораторну роботу ... 3
Теоретичні відомості ... 4
Практична частина ... 7
Опис програмної реалізації ... 7
Тестування реалізації на прикладі, наданому в умові лабораторної роботи ... 7
Множина Парето ... 8
Скаляризація ... 10
Ваги критеріїв ... 12
Зважені оцінки альтернатив ... 14
Комплексні оцінки альтернатив ... 14
Обчислення прикладу з умови без корекції третього критерія ... 16
Варіант 14 ... 18
Висновки ... 22
Перелік використаних джерел ... 23

Вступ

Мета

- вивчення методів і процедур багатокритеріального вибору альтернатив;
- застосування методів багатокритеріального вибору альтернатив для аналізу і вибору управлінських рішень.

Порядок виконання роботи

1. Ознайомитися з теоретичними відомостями з лабораторної роботи.
2. Отримати завдання на лабораторну роботу (див. Додаток).
3. Звести вихідні дані у таблицю.
4. Вибрати множину Парето.
5. Для альтернатив, що ввійшли до множини Парето, отримати безрозмірні оцінки.
6. Обрати найкращу альтернативу, використовуючи методику скаляризації векторних оцінок.
7. Зробити висновки

Завдання на лабораторну роботу

Запропонувати самостійно 10 (десять) альтернатив з параметрами. При цьому мати на увазі, що множина Парето має містити не менше 5-6 альтернатив.

14. Родина планує придбати ділянку під дачу. Вибір відбувається за наступними критеріями:
К1. Вартість (млн. грн.)
К2. Краса краєвиду (якісний критерій)
К3. Відстань від квартири (км)
К4. Площа ділянки (кв. м)
К5. Наявність під'їзних шляхів (якісний критерій)

Теоретичні відомості

1. Загальна характеристика задач і методів прийняття рішень при багатьох критеріях

У більшості випадків рішення (управлінські, проектно-технічні та інші) приймаються з урахуванням кількох критеріїв (цілей, показників якості). Тому більшість завдань, пов'язаних з прийняттям рішень, є багатокритеріальними. Інша назва таких завдань – завдання векторної оптимізації, оскільки рішення в них приймається з урахуванням набору з декількох критеріїв (т. з. вектором критеріїв). Якщо при виборі рішень враховується тільки один критерій, то таке завдання представляє собою завдання скалярної оптимізації. Прикладом завдання скалярної оптимізації може служити класична транспортна задача (завдання перевезення заданого кількості вантажів від постачальників споживачам з мінімальними витратами), оскільки в ній враховується лише один критерій – витрати на перевезення.

Серед задач прийняття рішень виділяють також дискретні і неперервні задачі. У дискретних задачах множина рішень (альтернатив) скінченна. Типові приклади – вибір одного з декількох товарів при покупці, вибір одного з можливих проектів будівництва підприємства і т. ін. У неперервних задачах є нескінченна множина можливих рішень (в межах деякого діапазону). Приклади таких задач – вибір оптимальних параметрів хімічної реакції при розробці технологічного процесу, вибір оптимального плану виробництва декількох сортів бензину і т. ін.

Очевидно, що задачі і скалярної, і векторної оптимізації можуть бути як неперервними, так і дискретними. Наприклад, якщо потрібно розподілити суму в розмірі 10 млн од. між декількома підприємствами, і ефективність вкладення коштів оцінюється за одним критерієм (наприклад, тільки за прибутком), то задача відноситься до класу задач скалярної оптимізації. Якщо враховується кілька критеріїв (наприклад, прибуток і термін окупності), то задача є задачею векторної оптимізації. Якщо кошти можуть виділятися тільки в розмірах, кратних одному мільйону, то дана задача – дискретна. Якщо кошти можуть виділятися в будь-яких (в межах наявної суми) кількостях, то задача відноситься до числа неперервних.

У даній роботі розглядаються методи і процедури розв'язання дискретних задач багатокритеріального вибору альтернатив. Такі задачі належать до слабоструктурованих і можуть розв'язуватися на основі методу аналізу ієрархій. Однак метод аналізу ієрархій вимагає великого обсягу експертної інформації: людина-експерт повинна виконати порівняння всіх критеріїв, а також всіх альтернатив за кожним з критеріїв. Легко підрахувати, що для вирішення завдання аналізу N альтернатив з використанням M критеріїв потрібно виконати $M \cdot (M-1)/2$ порівнянь критеріїв за важливістю і $M \cdot N \cdot (N-1)$ порівнянь альтернатив за критеріями. Таким чином, метод аналізу ієрархій досить трудомісткий. Крім того, цей метод має сенс застосовувати тільки за умови, що в рішенні завдання беруть участь висококваліфіковані фахівці (експерти), що не завжди можливо.

Тому розроблена низка простіших методів і процедур, що вимагають використання невеликого обсягу експертної інформації.

2. Вибір множини Парето

Вибір множини Парето-оптимальних рішень (множини Парето) являє собою відбір перспективних альтернатив, які явно кращі за інші.

Множина Парето – це множина альтернатив, які мають наступну властивість: будь-яка з альтернатив, що входять до множини Парето, хоча б за одним критерієм краща будь-якої іншої альтернативи з цієї множини. Іншими словами, жодна з альтернатив, що входять до множини Парето, не поступається якій-небудь альтернативі з цієї множини за всіма критеріями. Тому множину Парето називають ще множиною недомінованих альтернатив: в ній відсутні альтернативи, які явно (за всіма критеріями) відстають від якої-небудь іншої альтернативи.

Вибір множини Парето виконується наступним чином. *Всі* альтернативи *попарно* порівнюються *за всіма критеріями*. Якщо при порівнянні яких-небудь альтернатив (наприклад, A_i та A_j) виявляється, що одна з них (наприклад, A_j) *не краща за другу за жодним критерієм*, то її можна виключити з розгляду. Виключену альтернативу не потрібно порівнювати з іншими альтернативами, оскільки вона явно неперспективна.

Як правило, до множини Парето входять декілька альтернатив. Тому вибір множини Парето не забезпечує прийняття остаточного рішення (вибору одної найкращої альтернативи), проте дозволяє скоротити кількість альтернатив, що розглядаються.

3. Методика скаляризації векторних оцінок

Методика призначена для вибору раціональної альтернативи з множини альтернатив, які оцінюються за кількома критеріями.

Дана методика розрахована на вирішення завдань, в яких рішення приймається на основі числових критеріїв (або може бути виконаний перехід до таких критеріїв).

Основна перевага цієї методики – мінімальний обсяг інформації, яку потрібно отримати від ОПР або експерта для вибору рішення, що дозволяє практично повністю автоматизувати рішення задачі. У той же час недостатнє врахування суб'єктивних суджень ОПР є недоліком цієї методики.

Методика заснована на обчисленні узагальненої оцінки кожної альтернативи (з урахуванням оцінок за всіма критеріями) і зіставленні цих оцінок.

Методика реалізується в наступному порядку.

1. Оцінки альтернатив приводяться до числового безрозмірного вигляду. Для цього застосовуються такі процедури:

- для числових критеріїв, які підлягають максимізації, всі оцінки альтернатив за критерієм діляться на максимальну з оцінок за даним критерієм:
$$P_{ij} = \frac{X_{ij}}{\max\limits_{j} X_{ij}};$$
- для числових критеріїв, які підлягають мінімізації, з оцінок за даним критерієм вибирається мінімальна, і вона ділиться на всі оцінки альтернатив за даним критерієм:
$$P_{ij} = \frac{\min\limits_{j} X_{ij}}{X_{ij}};$$
- оцінки за якісними критеріями виражаються за п'ятибальною шкалою («відмінно», «добре», «задовільно», «погано», «дуже погано»), а потім виконується перехід до числових оцінок з використанням **шкали Харрінгтона**. При цьому оцінці «відмінно» відповідають числові оцінки від 0,8 до 1; «добре» – від 0,63 до 0,8; «задовільно» – від 0,37 до 0,63; «погано» – від 0,2 до 0,37; «дуже погано» – від 0 до 0,2. Числова оцінка виставляється людиною: експертом або особою, яка приймає рішення (ОПР). Наприклад, якщо за деяким критерієм дві альтернативи мають оцінку «добре», але одна з них дуже хороша, а інша – трохи гірша, то першій з альтернатив (кращій) можна призначити оцінку 0,8, а другий, наприклад, 0,7;
- для оцінок, що мають вид «так-ні» (тобто виражають наявність або відсутність деякого показника), зазвичай використовуються такі числові оцінки: «так» – 0,67, «ні» – 0,33, якщо оцінка «так» більш бажана, ніж «ні», «ні» – 0,67, «так» – 0,33, якщо більш бажаною є оцінка «ні».

В результаті переходу до безрозмірних оцінок усуваються відмінності вихідних оцінок, що утруднювали порівняння альтернатив. Безрозмірні величини не вимірюються в будь-яких одиницях, тому їх можна порівнювати одна з одною, додавати і т. д. Безрозмірні оцінки не розрізняються за діапазоном значень: всі вони мають значення в межах від 0 до 1. Вони не розрізняються також за спрямованістю: чим більше безрозмірна оцінка, тим краще (за будь-яким критерієм), і найкраще значення дорівнює 1.

2. Визначаються ваги (оцінки важливості) критеріїв. У даній методиці ваги знаходяться на основі розкиду оцінок. Ваги визначаються в такому порядку:

- визначаються середні оцінки за кожним критерієм:

$$\overline{P_i} = \frac{1}{N}\sum_{j=1}^{N} P_{ij}\,, i = 1,\ldots,M,$$

де M – кількість критеріїв, N – кількість альтернатив, P_{ij} – безрозмірні оцінки.

- знаходяться величини розкиду за кожним критерієм:

$$R_i = \frac{1}{N \cdot \overline{P_i}}\sum_{j=1}^{N} \left|P_{ij} - \overline{P_i}\right|, i = 1,\ldots,M.$$

- знаходиться сума величин розкиду:

$$R = \sum_{i=1}^{M} R_i.$$

- знаходяться ваги критеріїв, що відображають розкид оцінок:

$$V_i = \frac{R_i}{R}, i = 1,\ldots,M.$$

Чим більший розкид (тобто відмінність) в оцінках альтернатив за критерієм, тим більша вага цього критерію. Таким чином, критерії, за якими оцінки альтернатив істотно відрізняються, вважаються важливішими. Якщо оцінки альтернатив за якимось критерієм дуже близькі між собою, його вага буде мінімальною, оскільки порівняння альтернатив за близьких оцінок не має сенсу.

3. Знаходяться зважені оцінки альтернатив:

$$E_{ij} = \frac{V_i}{P_{ij}}, i = 1,\ldots,M, j = 1,\ldots,N.$$

Чим більших значень набувають безрозмірні оцінки P_{ij}, тим менші значення зважених оцінок. Таким чином, чим менші зважені оцінки, тим краща альтернатива.

4. Визначаються комплексні оцінки альтернатив (суми зважених оцінок):

$$E_j = \sum_{i=1}^{M} E_{ij}\,, j = 1,\ldots,N.$$

Чим ***менша*** комплексна оцінка, тим краща альтернатива.

Практична частина

Опис програмної реалізації

Метод багатокритеріального вибору, описаний в теоретичних відомостях, був реалізований мовою програмування С. Посилання на репозиторій (проект lab-4): https://github.com/Bohdan628318ylypchenko/system-analysis-labs.git

Тестування реалізації на прикладі, наданому в умові лабораторної роботи

Хімічний комбінат планує впровадити комплекс засобів автоматизації (КЗА) для системи управління технологічними процесами. Є можливість вибрати один з семи варіантів КЗА (КЗА1, КЗА2, ..., КЗА7). При виборі враховуються чотири критерії: К1 – витрати, пов'язані з виготовленням КЗА і його введенням в експлуатацію, млн грн.; К2 – термін введення КЗА в експлуатацію, міс.; К3 – термін гарантійного обслуговування підприємством-виробником, років; К4 – зручність КЗА в експлуатації. Характеристики КЗА наведені в таблиці 1.

Таблиця 1. Вихідні дані для прикладу 1

Показники	КЗА1	КЗА2	КЗА3	КЗА4	КЗА5	КЗА6	КЗА7
К1	40	30	40	60	45	25	55
К2	8	8	6	6	7	8	6
К3	4	4	5	7	4	4	5
К4	добре	відм.	задов.	відм.	погано	дуже добре	добре

```
long double ** data_2darr = (long double **)malloc(ALTERNATIVE_COUNT * sizeof(long double *));
NULL_EXIT(data_2darr);
for (int i = 0; i < ALTERNATIVE_COUNT; i++)
{
    data_2darr[i] = (long double *)malloc(CRITERIA_COUNT * sizeof(long double));
    NULL_EXIT(data_2darr[i]);
}

data_2darr[0][0] = 40.0; data_2darr[0][1] = 8.0; data_2darr[0][2] = 4.0; data_2darr[0][3] = 0.7;
data_2darr[1][0] = 30.0; data_2darr[1][1] = 8.0; data_2darr[1][2] = 4.0; data_2darr[1][3] = 1.0;
data_2darr[2][0] = 40.0; data_2darr[2][1] = 6.0; data_2darr[2][2] = 5.0; data_2darr[2][3] = 0.6;
data_2darr[3][0] = 60.0; data_2darr[3][1] = 6.0; data_2darr[3][2] = 7.0; data_2darr[3][3] = 0.9;
data_2darr[4][0] = 45.0; data_2darr[4][1] = 7.0; data_2darr[4][2] = 4.0; data_2darr[4][3] = 0.3;
data_2darr[5][0] = 25.0; data_2darr[5][1] = 8.0; data_2darr[5][2] = 4.0; data_2darr[5][3] = 0.8;
data_2darr[6][0] = 55.0; data_2darr[6][1] = 6.0; data_2darr[6][2] = 5.0; data_2darr[6][3] = 0.7;

criteria_type * criteria_type_1darr = (criteria_type *)malloc(CRITERIA_COUNT * sizeof(criteria_type));
NULL_EXIT(criteria_type_1darr);
criteria_type_1darr[0] = MIN;
criteria_type_1darr[1] = MIN;
criteria_type_1darr[2] = MAX;
criteria_type_1darr[3] = QUALITY;
```

Рис. 1 Задання вхідних даних у програмі (файл main.c)

Примітка: зазвичай зведення якісних оцінок до безрозмірних числових величин за допомогою шкали Харрінгтона виконується для множини Парето, не для початкової множини альтернатив. Таке зведення вимагає певної інтерактивності від програми: 1. отримати початкову множину 2. обрати множину Парето 3. Запросити нові

значення якісних критеріїв. З метою спрощення програми було вирішено, що на вхід програма прийматиме якісні критерії, які ВЖЕ ЗВЕДЕНІ ДО ЧИСЛОВИХ БЕЗРОЗМІРНИХ (так, маємо додаткове навантаження на експерта, адже експерт має дати ЧИСЛОВИЙ ЕКВІВАЛЕНТ якісних характеристики для ВСІХ альтернатив із початкової множини. Експерт був не проти (у нього не було вибору).

```
==========> Original data
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: U) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: U) = | 30.0000 | 8.0000 | 4.0000 | 1.0000 |
a 2 (PARETO SET FLAG: U) = | 40.0000 | 6.0000 | 5.0000 | 0.6000 |
a 3 (PARETO SET FLAG: U) = | 60.0000 | 6.0000 | 7.0000 | 0.9000 |
a 4 (PARETO SET FLAG: U) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: U) = | 25.0000 | 8.0000 | 4.0000 | 0.8000 |
a 6 (PARETO SET FLAG: U) = | 55.0000 | 6.0000 | 5.0000 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 0;
```

Введена початкова множина альтернатив

Множина Парето

Очікуваний результат:

Таблиця 2. Множина Парето для прикладу 1

Показники	КЗА2	КЗА3	КЗА4	КЗА6	КЗА7
К1	30	40	60	25	55
К2	8	6	6	8	6
К3	4	5	7	4	5
К4	відм.	задов.	відм.	дуже добре	добре

Результат роботи програми:

```
==========> Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 30.0000 | 8.0000 | 4.0000 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 40.0000 | 6.0000 | 5.0000 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 60.0000 | 6.0000 | 7.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 25.0000 | 8.0000 | 4.0000 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 55.0000 | 6.0000 | 5.0000 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
|1|2|3|5|6| (примітка: нумерація альтернатив починається з 0).
```

Результати співпадають.

Функції, що реалізують вибір множини Парето:

```c
void pareto_form_set(pareto_data * restrict const p_d)
{
    for (int i = 0; i < p_d->alternative_count; i++)
    {
        if (p_d->pareto_set[i] == EXCLUDED)
            continue;

        for (int j = 0; j < p_d->alternative_count; j++)
        {
            if ((i == j) || (p_d->pareto_set[j] == EXCLUDED))
                continue;

            if (_is_1st_better_by_all_criteria_2nd(p_d->criteria_count, p_d->criteria_type_1darr,
                                                   p_d->data_2darr[i], p_d->data_2darr[j]))
                p_d->pareto_set[j] = EXCLUDED;
        }
    }

    for (int i = 0; i < p_d->alternative_count; i++)
    {
        if (p_d->pareto_set[i] == EXCLUDED)
            continue;

        p_d->pareto_set[i] = INCLUDED;
        p_d->pareto_list[p_d->pareto_count++] = i;
    }
}
```

```c
static inline int _is_1st_better_by_all_criteria_2nd(int criteria_count,
                                                     const criteria_type * restrict const criteria_type_1darr,
                                                     const long double * restrict const a1,
                                                     const long double * restrict const a2)
{
    int score = 0;

    for (int i = 0; i < criteria_count; i++)
    {
        switch (criteria_type_1darr[i])
        {
        case BOOL:
        case QUALITY:
        case MAX:
            if (a1[i] >= a2[i])
                score++;
            else
                score--;
            break;
        case MIN:
            if (a1[i] <= a2[i])
                score++;
            else
                score--;
            break;
        default:
            INVALID_SWITCH_EXIT("invalid criteria type", criteria_type_1darr[i])
        }
    }

    return score == criteria_count;
}
```

(файл pareto_eval.c)

Скаляризація

Очікуваний результат:

Таблиця 3 Безрозмірні оцінки альтернатив для прикладу 1

Показники	КЗА2	КЗА3	КЗА4	КЗА6	КЗА7
К1	0,83	0,63	0,42	1	0,45
К2	0,75	1	1	0,75	1
К3	0,67	0,83	1	0,67	0,83
К4	1	0,6	0,9	0,8	0,7

Результат роботи програми:

```
==========> Normalized Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 0.8333 | 0.7500 | 0.5714 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 0.6250 | 1.0000 | 0.7143 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 0.4167 | 1.0000 | 1.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 1.0000 | 0.7500 | 0.5714 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 0.4545 | 1.0000 | 0.7143 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
|1|2|3|5|6|
```

Результати співпадають*, ЗА ВИНЯТКОМ ЗНАЧЕНЬ ТРЕТЬОГО КРИТЕРІЯ!*
Чому?

Коментар з прикладу:

«Критерій К3 підлягає максимізації. Тому всі оцінки за цим критерієм діляться на максимальну оцінку (в даному прикладі – на 7»

Розглянемо початкову множину Парето:

Показники	КЗА2	КЗА3	КЗА4	КЗА6	КЗА7
К1	30	40	60	25	55
К2	8	6	6	8	6
К3	4	5	7	4	5
К4	відм.	задов.	відм.	дуже добре	добре

Найбільше значення критерію №3 = 7.
Тоді нові значення мають бути: 4/7, 5/7, 1, 4/7, 5/7 = 0.5714, 0.7142, 1, 0.5714, 0.7142.
Тобто програма виконала обчислення правильно. Можливо в прикладі помилка?

Функції, що реалізують скаляризацію:

```c
void pareto_normalize(pareto_data * restrict const p_d)
{
    for (int i = 0; i < p_d->criteria_count; i++)
    {
        switch (p_d->criteria_type_1darr[i])
        {
        case BOOL:
            _normalize_bool(i, p_d);
            break;
        case MAX:
            _normalize_max(i, p_d);
            break;
        case MIN:
            _normalize_min(i, p_d);
            break;
        case QUALITY:
            break;
        default:
            INVALID_SWITCH_EXIT("invalid criteria type", p_d->criteria_type_1darr[i])
        }
    }
}
```

```c
static inline void _normalize_bool(int criteria_index, pareto_data * restrict const p_d)
{
    for (int i = 0; i < p_d->pareto_count; i++)
    {
        if (p_d->data_2darr[p_d->pareto_list[i]][criteria_index])
            p_d->data_2darr[p_d->pareto_list[i]][criteria_index] = NUMERIC_TRUE;
        else
            p_d->data_2darr[p_d->pareto_list[i]][criteria_index] = NUMERIC_FALSE;
    }
}

static inline void _normalize_max(int criteria_index, pareto_data * restrict const p_d)
{
    long double max = LDBL_MIN;

    for (int i = 0; i < p_d->pareto_count; i++)
        if (max < p_d->data_2darr[p_d->pareto_list[i]][criteria_index])
            max = p_d->data_2darr[p_d->pareto_list[i]][criteria_index];

    if (max == 0.0)
        return;

    for (int i = 0; i < p_d->pareto_count; i++)
        p_d->data_2darr[p_d->pareto_list[i]][criteria_index] /= max;
}

static inline void _normalize_min(int criteria_index, pareto_data * restrict const p_d)
{
    long double min = LDBL_MAX;

    for (int i = 0; i < p_d->pareto_count; i++)
        if (min > p_d->data_2darr[p_d->pareto_list[i]][criteria_index])
            min = p_d->data_2darr[p_d->pareto_list[i]][criteria_index];

    for (int i = 0; i < p_d->pareto_count; i++)
        if (p_d->data_2darr[p_d->pareto_list[i]][criteria_index] != 0)
            p_d->data_2darr[p_d->pareto_list[i]][criteria_index] =
            min / p_d->data_2darr[p_d->pareto_list[i]][criteria_index];
}
```

(файл pareto_eval.c)

Для подальшого тестування програми було вирішено вручну скорегувати безрозмірні числові значення для третього критерію:

```
48      puts("\n==========> Original data");
49      pareto_data_print(p_d);
50
51      pareto_form_set(p_d);
52      puts("\n==========> Pareto set");
53      pareto_data_print(p_d);
54
55      pareto_normalize(p_d);
56
57      p_d->data_2darr[1][2] = 0.67;
58      p_d->data_2darr[2][2] = 0.83;
59      p_d->data_2darr[3][2] = 1.0;
60      p_d->data_2darr[5][2] = 0.67;
61      p_d->data_2darr[6][2] = 0.83;
62
63      puts("\n==========> Normalized Pareto set");
64      pareto_data_print(p_d);
```

Обчислення без ручного корегування наведено в кінці цього розділу.

Ваги критеріїв

Очікувані результати:

$\overline{P}_1 = (0{,}83 + 0{,}63 + 0{,}42 + 1 + 0{,}45)/5 = 0{,}67,$
$\overline{P}_2 = (0{,}75 + 1 + 1 + 0{,}75 + 1)/5 = 0{,}9,$
$\overline{P}_3 = (0{,}67 + 0{,}83 + 1 + 0{,}67 + 0{,}83)/5 = 0{,}8,$
$\overline{P}_4 = (1 + 0{,}6 + 0{,}9 + 0{,}8 + 0{,}7)/5 = 0{,}8.$

- знаходимо величини розкиду за кожним критерієм:

$$R_1 = \frac{|0{,}83 - 0{,}67| + |0{,}63 - 0{,}67| + |0{,}42 - 0{,}67| + |1 - 0{,}67| + |0{,}45 - 0{,}67|}{5 \cdot 0{,}67} = 0{,}3,$$

$$R_2 = \frac{|0{,}75 - 0{,}9| + |1 - 0{,}9| + |1 - 0{,}9| + |0{,}75 - 0{,}9| + |1 - 0{,}9|}{5 \cdot 0{,}9} = 0{,}13,$$

$$R_3 = \frac{|0{,}67 - 0{,}8| + |0{,}83 - 0{,}8| + |1 - 0{,}8| + |0{,}67 - 0{,}8| + |0{,}83 - 0{,}8|}{5 \cdot 0{,}8} = 0{,}13,$$

$$R_4 = \frac{|1 - 0{,}8| + |0{,}6 - 0{,}8| + |0{,}9 - 0{,}8| + |0{,}8 - 0{,}8| + |0{,}7 - 0{,}8|}{5 \cdot 0{,}8} = 0{,}15;$$

- знаходимо суму величин розкиду:

$R = 0{,}3 + 0{,}13 + 0{,}13 + 0{,}15 = 0{,}71;$

- знаходимо ваги критеріїв:

$V_1 = 0{,}3/0{,}71 = 0{,}42,\ V_2 = 0{,}13/0{,}71 = 0{,}18,\ V_3 = 0{,}13/0{,}71 = 0{,}18,\ V_4 = 0{,}15/0{,}71 = 0{,}21.$

Результати роботи програми:

```
==========> Pareto P
| 0.6659 | 0.9000 | 0.8000 | 0.8000 |

==========> Pareto R
| 0.3013 | 0.1333 | 0.1300 | 0.1500 |

=========> Sum R: 0.714585

==========> Pareto v
| 0.4216 | 0.1866 | 0.1819 | 0.2099 |
```

Результати співпадають.

```c
void pareto_p(const pareto_data * restrict const p_d, long double * restrict const p)
{
    for (int i = 0; i < p_d->criteria_count; i++)
    {
        p[i] = 0.0;
        for (int j = 0; j < p_d->pareto_count; j++)
            p[i] += p_d->data_2darr[p_d->pareto_list[j]][i];
        p[i] /= p_d->pareto_count;
    }
}

void pareto_r(const pareto_data * restrict const p_d, const long double * restrict const p,
              long double * restrict const r)
{
    for (int i = 0; i < p_d->criteria_count; i++)
    {
        r[i] = 0.0;
        for (int j = 0; j < p_d->pareto_count; j++)
        {
            r[i] += fabsl(p_d->data_2darr[p_d->pareto_list[j]][i] - p[i]);
        }
        r[i] /= ((long double)p_d->pareto_count * p[i]);
    }
}

long double pareto_R(const pareto_data * restrict const p_d,
                     const long double * restrict const r)
{
    long double R = 0.0;
    for (int i = 0; i < p_d->criteria_count; i++)
        R += r[i];
    return R;
}

void pareto_v(const pareto_data * restrict const p_d,
              const long double * restrict const r, long double R,
              long double * restrict const v)
{
    for (int i = 0; i < p_d->criteria_count; i++)
        v[i] = r[i] / R;
}
```

Функції, що реалізують обчислення вагів (файл pareto_eval.c)

Зважені оцінки альтернатив

Очікуваний результат:

$$E_{11} = \frac{0,42}{0,83} = 0,51,\ E_{12} = \frac{0,42}{0,63} = 0,67,\ E_{13} = \frac{0,42}{0,42} = 1,\ E_{14} = \frac{0,42}{1} = 0,42,\ E_{15} = \frac{0,42}{0,45} = 0,93;$$

$$E_{21} = \frac{0,18}{0,75} = 0,24,\ E_{22} = \frac{0,18}{1} = 0,18,\ E_{23} = \frac{0,18}{1} = 0,18,\ E_{24} = \frac{0,18}{0,75} = 0,24,\ E_{25} = \frac{0,18}{1} = 0,18;$$

$$E_{31} = \frac{0,18}{0,67} = 0,27,\ E_{32} = \frac{0,18}{0,83} = 0,22,\ E_{33} = \frac{0,18}{1} = 0,18,\ E_{34} = \frac{0,18}{0,67} = 0,27,\ E_{35} = \frac{0,18}{0,83} = 0,22;$$

$$E_{41} = \frac{0,21}{1} = 0,21,\ E_{42} = \frac{0,21}{0,6} = 0,35,\ E_{43} = \frac{0,21}{0,9} = 0,23,\ E_{43} = \frac{0,21}{0,8} = 0,26,\ E_{43} = \frac{0,21}{0,7} = 0,3.$$

Результат роботи програми:

```
==========> Pareto e
pareto alternative 0 e matrix = | 0.5059 | 0.2488 | 0.2715 | 0.2099 |
pareto alternative 1 e matrix = | 0.6745 | 0.1866 | 0.2192 | 0.3499 |
pareto alternative 2 e matrix = | 1.0118 | 0.1866 | 0.1819 | 0.2332 |
pareto alternative 3 e matrix = | 0.4216 | 0.2488 | 0.2715 | 0.2624 |
pareto alternative 4 e matrix = | 0.9275 | 0.1866 | 0.2192 | 0.2999 |
```

0.4216 / 0.4167

Результати співпадають.

```
109  void pareto_ematrix(const pareto_data * restrict const p_d,
110                      const long double * restrict const v, const long double * restrict const p,
111                      long double * restrict const * restrict const eM)
112  {
113      for (int i = 0; i < p_d->pareto_count; i++)
114          for (int j = 0; j < p_d->criteria_count; j++)
115              eM[i][j] = v[j] / p_d->data_2darr[p_d->pareto_list[i]][j];
116  }
```

Функція, що реалізує обчислення зважених оцінок (файл pareto_eval.c)

Комплексні оцінки альтернатив

Очікувані результати:

$E_1 = 0,51 + 0,24 + 0,27 + 0,21 = 1,23,$
$E_2 = 0,67 + 0,18 + 0,22 + 0,35 = 1,42,$
$E_3 = 1 + 0,18 + 0,18 + 0,23 = 1,59,$
$E_4 = 0,42 + 0,24 + 0,27 + 0,26 = 1,19,$
$E_5 = 0,93 + 0,18 + 0,22 + 0,3 = 1,63,$

Найкращим є варіант КЗА6, на другому місці – КЗА2, на третьому – КЗА3, на четвертому – КЗА4, на п’ятому – КЗА7.

Результати роботи програми:

```
==========> Pareto SUM e
| 1.2361 | 1.4301 | 1.6135 | 1.2043 | 1.6331 |
==========>Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 0.8333 | 0.7500 | 0.6700 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 0.6250 | 1.0000 | 0.8300 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 0.4167 | 1.0000 | 1.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 1.0000 | 0.7500 | 0.6700 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 0.4545 | 1.0000 | 0.8300 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
|1|2|3|5|6|
```

Результати співпадають.

```
118  void pareto_earr(const pareto_data * restrict const p_d,
119                   const long double * restrict const * restrict const eM,
120                   long double * restrict const eA)
121  {
122      for (int i = 0; i < p_d->pareto_count; i++)
123      {
124          eA[i] = 0.0;
125          for (int j = 0; j < p_d->criteria_count; j++)
126              eA[i] += eM[i][j];
127      }
128  }
129
```

Функція, що реалізує обчислення комплексних оцінок

Обчислення прикладу з умови без корекції третього критерія

```
==========> Original data
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: U) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: U) = | 30.0000 | 8.0000 | 4.0000 | 1.0000 |
a 2 (PARETO SET FLAG: U) = | 40.0000 | 6.0000 | 5.0000 | 0.6000 |
a 3 (PARETO SET FLAG: U) = | 60.0000 | 6.0000 | 7.0000 | 0.9000 |
a 4 (PARETO SET FLAG: U) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: U) = | 25.0000 | 8.0000 | 4.0000 | 0.8000 |
a 6 (PARETO SET FLAG: U) = | 55.0000 | 6.0000 | 5.0000 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 0;

==========> Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 30.0000 | 8.0000 | 4.0000 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 40.0000 | 6.0000 | 5.0000 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 60.0000 | 6.0000 | 7.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 25.0000 | 8.0000 | 4.0000 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 55.0000 | 6.0000 | 5.0000 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
|1|2|3|5|6|

==========> Normalized Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 0.8333 | 0.7500 | 0.5714 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 0.6250 | 1.0000 | 0.7143 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 0.4167 | 1.0000 | 1.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 1.0000 | 0.7500 | 0.5714 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 0.4545 | 1.0000 | 0.7143 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
```

```
|1|2|3|5|6|

==========> Pareto P
| 0.6659 | 0.9000 | 0.7143 | 0.8000 |

==========> Pareto R
| 0.3013 | 0.1333 | 0.1600 | 0.1500 |

==========> Sum R: 0.744585

==========> Pareto v
| 0.4046 | 0.1791 | 0.2149 | 0.2015 |

==========> Pareto e
pareto alternative 0 e matrix = | 0.4855 | 0.2388 | 0.3760 | 0.2015 |
pareto alternative 1 e matrix = | 0.6473 | 0.1791 | 0.3008 | 0.3358 |
pareto alternative 2 e matrix = | 0.9710 | 0.1791 | 0.2149 | 0.2238 |
pareto alternative 3 e matrix = | 0.4046 | 0.2388 | 0.3760 | 0.2518 |
pareto alternative 4 e matrix = | 0.8901 | 0.1791 | 0.3008 | 0.2878 |


==========> RESULT

==========> Pareto SUM e
| 1.3018 | 1.4630 | 1.5888 | 1.2712 | 1.6578 |
==========>Pareto set
alternative count (row): 7; criteria count (column): 4;
a 0 (PARETO SET FLAG: N) = | 40.0000 | 8.0000 | 4.0000 | 0.7000 |
a 1 (PARETO SET FLAG: Y) = | 0.8333 | 0.7500 | 0.5714 | 1.0000 |
a 2 (PARETO SET FLAG: Y) = | 0.6250 | 1.0000 | 0.7143 | 0.6000 |
a 3 (PARETO SET FLAG: Y) = | 0.4167 | 1.0000 | 1.0000 | 0.9000 |
a 4 (PARETO SET FLAG: N) = | 45.0000 | 7.0000 | 4.0000 | 0.3000 |
a 5 (PARETO SET FLAG: Y) = | 1.0000 | 0.7500 | 0.5714 | 0.8000 |
a 6 (PARETO SET FLAG: Y) = | 0.4545 | 1.0000 | 0.7143 | 0.7000 |
criteria 0 = min
criteria 1 = min
criteria 2 = max
criteria 3 = quality
pareto count: 5;
|1|2|3|5|6|
```

Варіант 14

14. Родина планує придбати ділянку під дачу. Вибір відбувається за наступними критеріями:
К1. Вартість (млн. грн.)
К2. Краса краєвиду (якісний критерій)
К3. Відстань від квартири (км)
К4. Площа ділянки (кв. м)
К5. Наявність під'їзних шляхів (якісний критерій)

«Запропонувати самостійно 10 (десять) альтернатив з параметрами. При цьому мати на увазі, що множина Парето має містити не менше 5-6 альтернатив.»

Класифікація критеріїв:
К1. Вартість (млн. грн.): MIN
К2. Краса краєвиду (якісний критерій): QUALITY
К3. Відстань від квартири (км): MIN
К4. Площа ділянки (кв. м): MAX
К5. Наявність під'їзних шляхів (якісний критерій): QUALITY

10 альтернатив:

	К1 (MIN)	К2 (QUALITY)	К3 (MIN)	К4 (MAX)	К5(QUALITY)
A1	4	Відмінно (0.95)	35	150	Добре (0.75)
A2	4	Дуже погано (0.1)	72	300	Погано (0.34)
A3	3	Добре (0.79)	31	190	Добре (0.65)
A4	2	Погано (0.32)	24	224	Відмінно (0.94)
A5	5	Задовільно (0.45)	88	120	Дуже погано (0.15)
A6	8	Відмінно (0.97)	44	250	Добре (0.64)
A7	3	Відмінно (1)	29	150	Добре (0.79)
A8	7	Погано (0.25)	115	140	Задовільно (0.4)
A9	4	Добре (0.78)	40	220	Відмінно (0.88)
A10	6	Задовільно (0.51)	100	100	Задовільно (0.38)

Задання параметрів у програмі:

```
data_2darr[0][0] = 4.0; data_2darr[0][1] = 0.95; data_2darr[0][2] = 35.0 ; data_2darr[0][3] = 150.0; data_2darr[0][4] = 0.75;
data_2darr[1][0] = 4.0; data_2darr[1][1] = 0.10; data_2darr[1][2] = 72.0 ; data_2darr[1][3] = 300.0; data_2darr[1][4] = 0.34;
data_2darr[2][0] = 3.0; data_2darr[2][1] = 0.79; data_2darr[2][2] = 31.0 ; data_2darr[2][3] = 190.0; data_2darr[2][4] = 0.65;
data_2darr[3][0] = 2.0; data_2darr[3][1] = 0.32; data_2darr[3][2] = 24.0 ; data_2darr[3][3] = 224.0; data_2darr[3][4] = 0.94;
data_2darr[4][0] = 5.0; data_2darr[4][1] = 0.45; data_2darr[4][2] = 88.0 ; data_2darr[4][3] = 120.0; data_2darr[4][4] = 0.15;
data_2darr[5][0] = 8.0; data_2darr[5][1] = 0.97; data_2darr[5][2] = 44.0 ; data_2darr[5][3] = 250.0; data_2darr[5][4] = 0.64;
data_2darr[6][0] = 3.0; data_2darr[6][1] = 1.00; data_2darr[6][2] = 29.0 ; data_2darr[6][3] = 150.0; data_2darr[6][4] = 0.79;
data_2darr[7][0] = 7.0; data_2darr[7][1] = 0.25; data_2darr[7][2] = 115.0; data_2darr[7][3] = 140.0; data_2darr[7][4] = 0.40;
data_2darr[8][0] = 4.0; data_2darr[8][1] = 0.78; data_2darr[8][2] = 40.0 ; data_2darr[8][3] = 220.0; data_2darr[8][4] = 0.88;
data_2darr[9][0] = 6.0; data_2darr[9][1] = 0.10; data_2darr[9][2] = 72.0 ; data_2darr[9][3] = 165.0; data_2darr[9][4] = 0.34;
data_2darr[9][0] = 6.0; data_2darr[9][1] = 0.51; data_2darr[9][2] = 100.0; data_2darr[9][3] = 100.0; data_2darr[9][4] = 0.38;
```

```
criteria_type_1darr[0] = MIN;
criteria_type_1darr[1] = QUALITY;
criteria_type_1darr[2] = MIN;
criteria_type_1darr[3] = MAX;
criteria_type_1darr[4] = QUALITY;
```

Обчислення:

```
==========> Original data
alternative count (row): 10; criteria count (column): 5;
a 0 (PARETO SET FLAG: U) = | 4.0000 | 0.9500 | 35.0000 | 150.0000 | 0.7500 |
a 1 (PARETO SET FLAG: U) = | 4.0000 | 0.1000 | 72.0000 | 300.0000 | 0.3400 |
a 2 (PARETO SET FLAG: U) = | 3.0000 | 0.7900 | 31.0000 | 190.0000 | 0.6500 |
a 3 (PARETO SET FLAG: U) = | 2.0000 | 0.3200 | 24.0000 | 224.0000 | 0.9400 |
a 4 (PARETO SET FLAG: U) = | 5.0000 | 0.4500 | 88.0000 | 120.0000 | 0.1500 |
a 5 (PARETO SET FLAG: U) = | 8.0000 | 0.9700 | 44.0000 | 250.0000 | 0.6400 |
a 6 (PARETO SET FLAG: U) = | 3.0000 | 1.0000 | 29.0000 | 150.0000 | 0.7900 |
a 7 (PARETO SET FLAG: U) = | 7.0000 | 0.2500 | 115.0000 | 140.0000 | 0.4000 |
a 8 (PARETO SET FLAG: U) = | 4.0000 | 0.7800 | 40.0000 | 220.0000 | 0.8800 |
a 9 (PARETO SET FLAG: U) = | 6.0000 | 0.5100 | 100.0000 | 100.0000 | 0.3800 |
criteria 0 = min
criteria 1 = quality
criteria 2 = min
criteria 3 = max
criteria 4 = quality
pareto count: 0;

==========> Pareto set
alternative count (row): 10; criteria count (column): 5;
a 0 (PARETO SET FLAG: N) = | 4.0000 | 0.9500 | 35.0000 | 150.0000 | 0.7500 |
a 1 (PARETO SET FLAG: Y) = | 4.0000 | 0.1000 | 72.0000 | 300.0000 | 0.3400 |
a 2 (PARETO SET FLAG: Y) = | 3.0000 | 0.7900 | 31.0000 | 190.0000 | 0.6500 |
a 3 (PARETO SET FLAG: Y) = | 2.0000 | 0.3200 | 24.0000 | 224.0000 | 0.9400 |
a 4 (PARETO SET FLAG: N) = | 5.0000 | 0.4500 | 88.0000 | 120.0000 | 0.1500 |
a 5 (PARETO SET FLAG: Y) = | 8.0000 | 0.9700 | 44.0000 | 250.0000 | 0.6400 |
a 6 (PARETO SET FLAG: Y) = | 3.0000 | 1.0000 | 29.0000 | 150.0000 | 0.7900 |
a 7 (PARETO SET FLAG: N) = | 7.0000 | 0.2500 | 115.0000 | 140.0000 | 0.4000 |
a 8 (PARETO SET FLAG: Y) = | 4.0000 | 0.7800 | 40.0000 | 220.0000 | 0.8800 |
a 9 (PARETO SET FLAG: N) = | 6.0000 | 0.5100 | 100.0000 | 100.0000 | 0.3800 |
criteria 0 = min
criteria 1 = quality
```

```
criteria 2 = min
criteria 3 = max
criteria 4 = quality
pareto count: 6;
|1|2|3|5|6|8|

==========> Normalized Pareto set
alternative count (row): 10; criteria count (column): 5;
a 0 (PARETO SET FLAG: N) = | 4.0000 | 0.9500 | 35.0000 | 150.0000 | 0.7500 |
a 1 (PARETO SET FLAG: Y) = | 0.5000 | 0.1000 | 0.3333 | 1.0000 | 0.3400 |
a 2 (PARETO SET FLAG: Y) = | 0.6667 | 0.7900 | 0.7742 | 0.6333 | 0.6500 |
a 3 (PARETO SET FLAG: Y) = | 1.0000 | 0.3200 | 1.0000 | 0.7467 | 0.9400 |
a 4 (PARETO SET FLAG: N) = | 5.0000 | 0.4500 | 88.0000 | 120.0000 | 0.1500 |
a 5 (PARETO SET FLAG: Y) = | 0.2500 | 0.9700 | 0.5455 | 0.8333 | 0.6400 |
a 6 (PARETO SET FLAG: Y) = | 0.6667 | 1.0000 | 0.8276 | 0.5000 | 0.7900 |
a 7 (PARETO SET FLAG: N) = | 7.0000 | 0.2500 | 115.0000 | 140.0000 | 0.4000 |
a 8 (PARETO SET FLAG: Y) = | 0.5000 | 0.7800 | 0.6000 | 0.7333 | 0.8800 |
a 9 (PARETO SET FLAG: N) = | 6.0000 | 0.5100 | 100.0000 | 100.0000 | 0.3800 |
criteria 0 = min
criteria 1 = quality
criteria 2 = min
criteria 3 = max
criteria 4 = quality
pareto count: 6;
|1|2|3|5|6|8|

==========> Pareto P
| 0.5972 | 0.6600 | 0.6801 | 0.7411 | 0.7067 |

==========> Pareto R
| 0.3023 | 0.4545 | 0.2752 | 0.1604 | 0.2311 |

==========> Sum R: 1.423628

==========> Pareto v
| 0.2124 | 0.3193 | 0.1933 | 0.1127 | 0.1624 |

==========> Pareto e
pareto alternative 0 e matrix = | 0.4247 | 3.1929 | 0.5799 | 0.1127 | 0.4775 |
pareto alternative 1 e matrix = | 0.3185 | 0.4042 | 0.2497 | 0.1779 | 0.2498 |
pareto alternative 2 e matrix = | 0.2124 | 0.9978 | 0.1933 | 0.1509 | 0.1727 |
pareto alternative 3 e matrix = | 0.8495 | 0.3292 | 0.3544 | 0.1352 | 0.2537 |
pareto alternative 4 e matrix = | 0.3185 | 0.3193 | 0.2336 | 0.2254 | 0.2055 |
pareto alternative 5 e matrix = | 0.4247 | 0.4093 | 0.3222 | 0.1537 | 0.1845 |


==========> RESULT
```

```
==========> Pareto SUM e
| 4.7877 | 1.4001 | 1.7271 | 1.9219 | 1.3023 | 1.4944 |

==========>Pareto set
alternative count (row): 10; criteria count (column): 5;
a 0 (PARETO SET FLAG: N) = | 4.0000 | 0.9500 | 35.0000 | 150.0000 | 0.7500 |
a 1 (PARETO SET FLAG: Y) = | 0.5000 | 0.1000 | 0.3333 | 1.0000 | 0.3400 |
a 2 (PARETO SET FLAG: Y) = | 0.6667 | 0.7900 | 0.7742 | 0.6333 | 0.6500 |
a 3 (PARETO SET FLAG: Y) = | 1.0000 | 0.3200 | 1.0000 | 0.7467 | 0.9400 |
a 4 (PARETO SET FLAG: N) = | 5.0000 | 0.4500 | 88.0000 | 120.0000 | 0.1500 |
a 5 (PARETO SET FLAG: Y) = | 0.2500 | 0.9700 | 0.5455 | 0.8333 | 0.6400 |
a 6 (PARETO SET FLAG: Y) = | 0.6667 | 1.0000 | 0.8276 | 0.5000 | 0.7900 |
a 7 (PARETO SET FLAG: N) = | 7.0000 | 0.2500 | 115.0000 | 140.0000 | 0.4000 |
a 8 (PARETO SET FLAG: Y) = | 0.5000 | 0.7800 | 0.6000 | 0.7333 | 0.8800 |
a 9 (PARETO SET FLAG: N) = | 6.0000 | 0.5100 | 100.0000 | 100.0000 | 0.3800 |
criteria 0 = min
criteria 1 = quality
criteria 2 = min
criteria 3 = max
criteria 4 = quality
pareto count: 6;
|1|2|3|5|6|8|
```

Множина Парето налічує 6 елементів.
Найкращою альтернативою є A7(6 + 1), потім – A3(2 + 1), A9(8 + 1), A4(3+1), A6(5 + 1), A2(1 + 1).

Висновки

Під час виконання лабораторної роботі було досліджено методи вибору з урахуванням кількох критеріїв, їхнє використання на прикладі процедуру попарного порівняння альтернатив. Це дозволило нам оцінити кожну альтернативу за кількома критеріями одночасно, керуючись значенням кожного критерію для досягнення оптимального результату. Було вивчено принцип застосування методу Парето, який є важливим інструментом у багатокритеріальному аналізі у процесі управлінського прийняття рішень.

Топ-3 альтернатив із запропонованих:

1 – A7,
2 – A3,
3 – A9.

Перелік використаних джерел

1. Теоретичні матеріали надані до лабораторної роботи 4 з предмету “Системний аналіз”, тема “Побудова функціональної моделі системи”, Вовк Лілія Борисівна.