

Т. В. Теслюк, аспірант кафедри АСУ, НУ “Львівська політехніка”,  
А. Я. Зелінський, к.т.н., ст. викл. кафедри ІСТ, НУ “Львівська політехніка”,  
Л. Є. Угрин, ст. викл. кафедри ІСТ, НУ “Львівська політехніка”,  
В. Я. Коваль, асист. кафедри ІСТ, НУ “Львівська політехніка”.

## СИСТЕМА РОЗВ’ЯЗАННЯ ЗАДАЧ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ З ВИКОРИСТАННЯМ ПОБУДОВИ МНОЖИНИ ОПТИМАЛЬНИХ РІШЕНЬ – ПАРЕТО

**Abstract.** The software and information support of the multi-criteria optimization problem solving system (SCO) is developed on the basis of the construction of the Pareto set. The built-in software is based on the Java language, and the information is based on the developed data structures. The results of the work of the developed software are presented.

**Анотація.** В роботі розроблено програмне та інформаційне забезпечення системи розв’язання задач багатокритеріальної оптимізації (ЗБО) на основі побудови множини Парето. Побудоване програмне забезпечення ґрунтується на мові Java, а інформаційне – на основі розроблених структур даних. Наведено результати застосування розробленого програмного забезпечення.

### Вступ

В процесі синтезу технічних систем постає задача забезпечення параметрів технічного завдання. Такі задачі можна розв’язати використовуючи методи багатокритеріальної оптимізації [1]. Існують різні методи та підходи до розв’язання таких задач. Зокрема, можна використати методи та алгоритми, що передбачають розв’язання послідовності задач однокритеріальної оптимізації, а саме: метод головної компоненти, метод поступок, методи цільового програмування [2] та інші. Ці методи мають суттєвий недолік, який пов’язаний з суб’єктивізмом у визначенні важливості критеріїв оптимізації.

Інша група методів ґрунтується на побудові інтегрального критерію оптимізації [3]: адитивна та мультиплікативні згортки, принцип максіміна та мінімакса. На сьогодні, ці методи використовуються в процесі розроблення складних технічних систем, але застосовувати їх при великій кількості альтернатив – недоцільно. Оскільки, потребують величезних обчислювальних ресурсів комп’ютера. Значно ефективніше використати підхід, який передбачає на першому кроці визначення альтернатив, що належать до множини Парето [4], а на другому – використовуючи метод аналізу ієрархій [5, 6], метод ELECTRE чи методи з використанням інтегрального критерію. Тому, розроблення системи розв’язання задач багатокритеріальної оптимізації (ЗБО) на основі побудови множини Парето є актуальним завданням.

## 1. Розроблення структури системи розв'язання ЗБО

Перший етап розроблення програмної системи пов'язаний з побудовою її структури. Спроектвана структура системи розв'язання ЗБО на основі побудови множини Парето ґрунтується на модульному принципі і включає такі основні складові (див. рис.1): підсистема введення даних; підсистема контролю введених даних; модуль забезпечення інтерфейсу користувача; модуль обчислень; модуль вводу результатів обчислень у текстовому форматі; модуль представлення даних в XML-форматі.

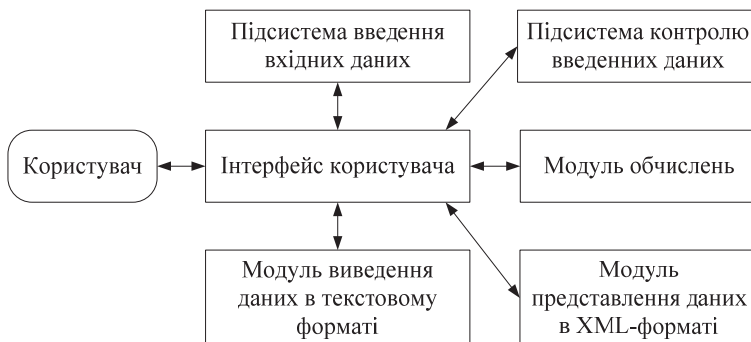


Рис. 1. Структура системи зменшення потужності множини альтернативних рішень з використанням критерію Парето

Модульна організація системи дає змогу швидко та ефективно вносити зміни в процесі її вдосконалення та розвитку.

В процесі розв'язання ЗБО, використано алгоритм знаходження рішень, що належать до множини Парето  $P(Y)$ , який має наступні кроки.

Крок 1. Присвоїти  $P(Y) = Y$ ,  $i = 1$ ,  $j = 2$ . Тим самим створити так звану початкову множину Парето, яка на початку роботи алгоритму співпадає з множиною  $Y$ , а в кінці – сформує шукану множину Парето – оптимальних рішень багатокритеріальної задачі. Алгоритм побудований таким чином, що шукана множина Парето формується з  $Y$  послідовним видаленням свідомо неоптимальних векторів.

Крок 2. Перевірити виконання нерівності  $y_i \geq y_j$ . Якщо нерівність виявилася істинною, то перейти до Кроку 3. В іншому випадку перейти до Кроку 5.

Крок 3. Видалити з поточної множини векторів  $P(Y) = Y$  вектор  $y_j$ , так як він не є Парето-оптимальним. Потім перейти до Кроку 4.

Крок 4. Перевірити виконання нерівності  $j < N$ . Якщо вона виконується, то присвоїти  $j = j + 1$  і повернутися до Кроку 2. В іншому

випадку – перейти до Кроку 7.

Крок 5. Перевірити правдивість нерівності  $y_j \geq y_i$ . У тому випадку, коли вона є істинною, перейти до Кроку 6. В іншому випадку – повернутися до Кроку 4.

Крок 6. Видалити з поточної множини векторів  $P(Y) = Y$  вектор  $y_i$  і перейти до Кроку 7.

Крок 7. Перевірити виконання нерівності  $i < N - 1$ . У разі істинності цієї нерівності слід послідовно присвоїти  $i = i + 1$ , а потім  $j = i + 1$ . Після цього необхідно повернутися до Кроку 2. В іншому випадку (тобто коли  $i < N - 1$ ) обчислення закінчити.

## 2. Розроблення програмного забезпечення системи

Для реалізації системи, використана мова Java. Розроблена та реалізована діаграма класів програмного забезпечення зображена на рис.2, а призначення кожного кроку – в таблиці 1.



Рис.2. Пакети класів розробленої системи

Таблиця 1

Структура пакетів аплікації

Пакет	Призначення	Класи
org.lp.qs	Кореневий пакет аплікації, містить клас з точкою входу	App.java
org.lp.qs.ui	Пакет містить клас основного фрейму	MainFrame.java
org.lp.qs.model.api	Пакет містить клас параметрів та інтерфейс системи	Parameter.java Decision.java
org.lp.qs.pareto.impl	Пакет містить класи реалізації методу Парето	ParetoObject.java

Програмне забезпечення системи написано на мові програмування JAVA з використанням стандартних бібліотек та бібліотеки JAVA FX для розробки інтерфейсу користувача.

Система забезпечує користувача зручним інтерфейсом та дає змогу розв'язувати задачі великої розмірності. Розроблений додаток дає змогу завантажити файл вхідних даних з диску на комп'ютері, обчислити Парето – оптимальні рішення та записати отримані значення у файл.

Основним класом програми є клас `ParetoObject`, який містить всі поля та методи, які необхідні для вдалого обчислення Парето – оптимальної множини рішень. Клас оперує даними класу `Decision`, який представляє одне рішення, і містить два поля `int id` – порядковий номер рішення, `double[] data` – масив значень критеріїв. Також розроблено декілька класів контролерів для користувальницького інтерфейсу з бібліотеки `JavaFX`, в яких містяться методи, які оперують даними, зв'язують дані з відповідними об'єктами користувальницького інтерфейсу, та методи зчитування з файлу та запису даних в файл. Отже, використання мови `Java` забезпечує платформонезалежність розробленого програмного продукту

### 3. Розроблення інформаційного забезпечення

Розроблене інформаційне забезпечення системи ґрунтується на структурах даних вхідного та вихідного файлів. Програмна система дає змогу зчитати в програму вхідні дані з використанням спеціального меню та вхідного і вихідного файлів. В цьому випадку, система зчитує вхідні дані з файлу з довільним іменем і розширенням `*.TXT` та структури даних, приклад якої зображено на рис.3.

В цьому файлі повинні міститися опції, такі як значення за яким будуть проводитися порівняння (може бути `max` або `min`) після мітки `#EVALUATE_BY`, кількість рішень в файлі після мітки `#DECISIONS_COUNT`, кількість критеріїв за якими проводяться порівняння після мітки `#CRITERIONS_COUNT`, та проіндексованих даних після мітки `#DATA`.

```
#EVALUATE_BY
max
#DECISIONS_COUNT
20
#CRITERIONS_COUNT
3
#DATA
0)72.06;34.23;66.01;
1)49.31;42.22;95.34;
2)16.66;72.43;39.4;
...
```

Рис.3. Структура вхідного файлу

Дані представляються десятковими дробовими числами у пронумерованих рядках, де стовпці розділені знаком «;» (крапка з комою),

кількість стовпців відповідає кількості критеріїв. Кожен рядок даних має починатися з цілого числа та символу «»), для індексації.

#### 4. Приклад розв’язаної ЗБО з використанням розробленої системи

Для розв’язання поставленої задачі з використанням розробленої системи необхідно завантажити систему на виконання. Введемо вхідні дані (рис. 4)

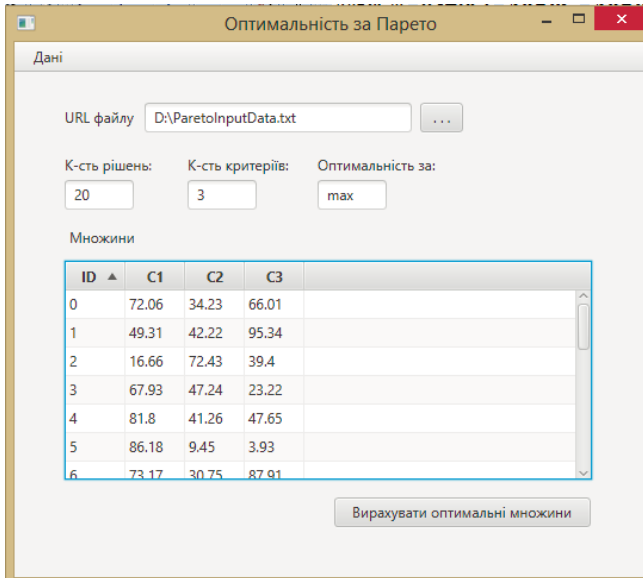


Рис. 4. Головне вікно після завантаження файлу

Після завантаження файлу, дані поміщаються в таблицю, де кожний рядок представляє собою рішення (стовпець ID – індекс рішення, а стовпці C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub> ... – критерії).

Для того щоб обчислити множину Парето оптимальних рішень, потрібно натиснути на кнопку «Вирахувати оптимальні множини» (рис. 4).

Розгорнуті результати можна переглянути в збереженому файлі на диску (рис.5).

Отже, розроблена система дає змогу зменшити потужність множини альтернативних рішень з використанням підходу на основі побудови множини оптимальних рішень Парето. Отримані результати тестування дають можливість стверджувати, що система працює правильно та коректно.

#### Висновки

Розроблено програмне забезпечення системи зменшення потужності множини альтернативних рішень на основі побудови множини оптимальних

рішень Парето. При реалізації системи використано технологію Java, що дає змогу використовувати програмний продукт під різними операційними системами та на різних технічних платформах.

Розроблено інформаційне забезпечення та структурну схему роботи системи. Наведено результати тестування розробленого програмного засобу, що дає змогу стверджувати про правильність та коректність роботи системи.

*Результати програми:*

*Програму було успішно виконано за 32 порівняння.*

*Кількість рішень: 20*

*Кількість критеріїв: 3*

*Оптимальність за: max*

*Кількість Парето - оптимальних рішень: 6*

*Парето - оптимальні множини:*

*1) {49.31; 42.22; 95.34}*

*6) {73.17; 30.75; 87.91}*

*7) {49.94; 95.35; 52.54}*

*.....*

*Відкинуті множини:*

*0) {72.06; 34.23; 66.01}*

*2) {16.66; 72.43; 39.4}*

*3) {67.93; 47.24; 23.22}*

Рис. 6. Результати тестової роботи програми

1. Denysyuk P., Teslyuk V., T. A. AlOmari, Teslyuk T. Development and study of subsystem for solution of tasks of multicriterial optimization // Proceedings of the 5th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2009, Lviv – Polyana, Ukraine, 2009, pp. 166–167.
2. Теслюк В.М., Загарюк Р.В. Методи багатокритеріальної оптимізації. Конспект лекцій з курсу «Методи багатокритеріальної оптимізації» для студентів базового напрямку 6.050101 «Комп'ютерні науки», Ч.1. Львів, 2012. - 52с.
3. Marler R.T. and Arora J.S. Survey of multi-objective optimization methods for engineering / Struct Multidisc Optim 26, 2004, P. 369–395.
4. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. - М.: Наука, 1982.
5. Лотов А. В., Бушенков В. А., Каменев Г. К., Черных О. Л. Компьютер и поиск компромисса. Метод достижимых целей. - М.: Наука, 1997.
6. Цмоць І.Г., Теслюк В. М., Теслюк Т.В. Модель визначення пріоритетності виконання інвестиційних енергозберігаючих проєктів на підприємстві // Актуальні проблеми економіки. – 2017. – № 1 (187). – Р. 398 – 408.

*Поступила 19.02.2018р.*