

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут заочно-дистанційного навчання
Форма навчання заочна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)
«__» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

**РОЗВ'ЯЗУВАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ З ДРОБОВО-ЛІНІЙНОЮ
ЦІЛЬОВОЮ ФУНКЦІЄЮ НА РОЗМІЩЕННЯХ**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістра

Виконавець роботи Черненко Оксана Олексіївна
_____ «__» _____ 2025р.
(підпис)

Науковий керівник к. ф.-м. н., доцент, Ольховський Дмитро Миколайович
_____ «__» _____ 2025 р.
(підпис)

Рецензент

ЗАТВЕРДЖУЮ
Завідувач кафедри _____ Олена ОЛЬХОВСЬКА
«__» _____ 2024 р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розв'язування задач оптимізації з дробово-лінійною цільовою функцією на розміщеннях»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістра

Прізвище, ім'я, по батькові Черненко Оксана Олексіївна

Затверджена наказом ректора № 211-Н від «30» жовтня 2023 р.

Термін подання студентом роботи «__» _____ 202_ р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, статті та документації з теми розробки програмного забезпечення, тестування, стандарти

ВСТУП

РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Розвиток теорії комбінаторної оптимізації

2.2. Огляд методів розв'язування комбінаторних задач

**РОЗДІЛ 3 МЕТОД РОЗВ'ЯЗУВАННЯ БЕЗУМОВНОЇ ЗАДАЧІ З
ДРОБОВО-ЛІНІЙНОЮ ФУНКЦІЄЮ ЦІЛІ НА РОЗМІЩЕННЯХ**

3.1. Постановка задачі

3.2. Розв'язування безумовної задачі з дробово-лінійною цільовою функцією на множині розміщень

РОЗДІЛ 4 МЕТОД РОЗВ'ЯЗУВАННЯ УМОВНИХ ЗАДАЧ З ДРОБОВО-ЛІНІЙНОЮ ЦІЛЬОВОЮ ФУНКЦІЄЮ НА ЗАГАЛЬНІЙ МНОЖИНІ РОЗМІЩЕНЬ

РОЗМІЩЕНЬ

4.1. Постановка задачі

4.2. Метод побудови лексикографічної еквівалентності для задач з дробово-лінійною цільовою функцією на загальній множині розміщень

4.3. Алгоритми, їх обґрунтування та оцінки

4.4. Числові експерименти та приклади за методом побудови лексикографічної еквівалентності

ВИСНОВКИ

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Ольховський Д.М.		

Інформаційний огляд	Ольховський Д.М.		
Теоретична частина	Ольховський Д.М.		
Практична частина	Ольховський Д.М.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 202__ р.

Здобувач вищої освіти _____ Черненко Оксана Олексіївна
(підпис)

Науковий керівник _____ к.ф.-м.н., доц. Ольховський Д.М.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена

на _____ (балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «__» _____ 202__ р.

Секретар ЕК _____ (підпис) _____ (ініціали та прізвище)

Затверджую
 Зав. кафедрою _____
 к.ф.-м.н. Олена ОЛЬХОВСЬКА
 «____» _____ 202__ р.

Погоджено
 Науковий керівник _____
 к.ф.-м.н. Дмитро ОЛЬХОВСЬКИЙ
 «____» _____ 202__ р.

План

кваліфікаційної роботи на тему «Розв'язування задач оптимізації з дробово-лінійною цільовою функцією на розміщеннях»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістра

Прізвище, ім'я, по батькові Черненко Оксана Олексіївна

ВСТУП

РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Розвиток теорії комбінаторної оптимізації

2.2. Огляд методів розв'язування комбінаторних задач

РОЗДІЛ 3 МЕТОД РОЗВ'ЯЗУВАННЯ БЕЗУМОВНОЇ ЗАДАЧІ З ДРОБОВО-ЛІНІЙНОЮ ФУНКЦІЄЮ ЦІЛІ НА РОЗМІЩЕННЯХ

3.1. Постановка задачі

3.2. Розв'язування безумовної задачі з дробово-лінійною цільовою функцією на множині розміщень

РОЗДІЛ 4 МЕТОД РОЗВ'ЯЗУВАННЯ УМОВНИХ ЗАДАЧ З ДРОБОВО-ЛІНІЙНОЮ ЦІЛЬОВОЮ ФУНКЦІЄЮ НА ЗАГАЛЬНІЙ МНОЖИНІ РОЗМІЩЕНЬ

4.1. Постановка задачі

4.2. Метод побудови лексикографічної еквівалентності для задач з дробово-лінійною цільовою функцією на загальній множині розміщень

4.3. Алгоритми, їх обґрунтування та оцінки

4.4. Числові експерименти та приклади за методом побудови лексикографічної еквівалентності

ВИСНОВКИ

Здобувач вищої освіти _____ Черненко Оксана
 «____» _____ 202__ р.

РЕФЕРАТ

Записка: 92 с., 4 рис., 6 таблиць, 26 джерел.

Предмет розробки – евклідові задачі комбінаторної оптимізації з дробово-лінійною цільовою функцією на розміщеннях, методи їх розв’язування.

Об’єкт розробки – комбінаторні задачі оптимізації та методи їх розв’язування.

Методи дослідження– методи евклідової комбінаторної оптимізації, а також методи математичного програмування (для одержання розв’язків задач оптимізації з дробово-лінійною функцією цілі на загальній множині розміщень).

Результати дослідження. Побудовано алгоритм розв’язування безумовних задач оптимізації з дробово-лінійними функціями цілі на розміщеннях на основі властивостей комбінаторного многогранника. Запропоновано алгоритм розв’язування умовних задач з дробово-лінійною функцією цілі на загальній множині розміщень, як подальший розвиток алгоритму розв’язування лінійних задач на розміщеннях. Проведені оцінки роботи допоміжних алгоритмів, складових алгоритму розв’язування задачі, та числові експерименти, які показали його практичну його ефективність.

Наукова апробація. За матеріалами роботи опубліковано статті у фахових наукових журналах.

Ключові слова: евклідова оптимізація, оцінки алгоритмів, збіжність, метод гілок та меж, симплекс-метод, комбінаторні методи.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ.....	12
РОЗДІЛ 2 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	13
2.1. Розвиток теорії комбінаторної оптимізації	13
2.2. Огляд методів розв’язування комбінаторних задач.....	15
РОЗДІЛ 3 МЕТОД РОЗВ’ЯЗУВАННЯ БЕЗУМОВНОЇ ЗАДАЧІ З ДРОБОВО-ЛІНІЙНОЮ ФУНКЦІЄЮ ЦІЛІ НА РОЗМІЩЕННЯХ.....	17
3.1. Постановка задачі.....	17
3.2. Розв’язування безумовної задачі з дробово-лінійною цільовою функцією на множині розміщень.....	21
РОЗДІЛ 4 МЕТОД РОЗВ’ЯЗУВАННЯ УМОВНИХ ЗАДАЧ З ДРОБОВО-ЛІНІЙНОЮ ЦІЛЬОВОЮ ФУНКЦІЄЮ НА ЗАГАЛЬНІЙ МНОЖИНІ РОЗМІЩЕНЬ.....	31
4.1. Постановка задачі.....	31
4.2. Метод побудови лексикографічної еквівалентності для задач з дробово-лінійною цільовою функцією на загальній множині розміщень.....	33
4.3. Алгоритми, їх обґрунтування та оцінки.....	39
4.4. Числові експерименти та приклади за методом побудови лексикографічної еквівалентності.....	61
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
$A_{\eta^n}^k(G)$	загальна множина розміщень по k елементів з мультимножини η елементів, серед яких n різних
$conv M$	опукла оболонка множини M
$E_{\eta^n}^k(G)$	образ загальної множини k -розміщень при зануренні в арифметичний евклідов простір R^k
$[G]$	первинна специфікація мультимножини G
J_k	множина перших k натуральних чисел
$k_G(e) = \eta$	кратність елемента e у мультимножині G
l -метод	лексикографічна модифікація двоїстого симплекс-методу
l -антинормальна симплекс-таблиця	симплекс-таблиця, в якій перша відмінна від нуля компонента довільного вектора R_j , де $j \in N_k$, є від'ємною
l -нормальна симплекс-таблиця	симплекс-таблиця, в якій перша відмінна від нуля компонента довільного вектора R_j , де $j \in N_k$, є додатньою
$ M $	кількість елементів у множині M
N_k	множина небазисних змінних на ітерації k
$\Pi_{\eta^n}^k(G)$	загальний многогранник розміщень
$Q_{\eta^n}^k(G)$	образ многогранника розміщень при переході від задачі з дробово-лінійною

	цільовою функцією до лінійної
R^n	n - вимірний евклідів простір
$\rho(\bar{V})$	ранг класу $\bar{V}$
$S(G)$	основа мультимножини G
$\lfloor V_1 \rfloor$	найближчий зліва до $\bar{V}$ комбінаторний клас з представником $\lfloor x_1 \rfloor$
$\lceil V_1 \rceil$	найближчий справа до $\bar{V}$ комбінаторний клас з представником $\lceil x_1 \rceil$

ВСТУП

Магістерська робота присвячена дослідженню властивостей математичних моделей евклідових комбінаторних задач оптимізації, а саме, властивостей многогранника – області допустимих розв’язків лінійних задач, до яких зводяться задачі на розміщеннях з дробово-лінійною цільовою функцією, а також розробці методів розв’язування цих задач.

Дослідження задач дискретної оптимізації є передумовою успішного моделювання важливих економічних, природних, соціальних та інших процесів. Велика кількість публікацій в області дискретної оптимізації [1-26] свідчить про необхідність та важливість подібних досліджень. В Україні серед вчених, роботи яких присвячені різним напрямам дискретної оптимізації, в першу чергу слід назвати Сергієнка І. В., Шора Н. З., Стояна Ю. Г., а також Гуляницького Л. Ф., Донця Г. П., Ємця О. О., Ляшенка І. М., Павлова О. А., Панішева А. В., Червака Ю. Ю., Шиля В. П., Яковлева С. В. та їх наукові колективи.

Важливий клас дискретних оптимізаційних задач складають оптимізаційні задачі комбінаторного типу, зокрема, задачі евклідової комбінаторної оптимізації.

У рамках дискретної оптимізації досліджуються задачі і з дробово-лінійною цільовою функцією. Необхідність адекватного моделювання оптимізаційними задачами, а саме, побудова моделі, що в повній мірі відображає реальні властивості задач (зокрема, бути розміщенням з деякої множини), викликає потребу розглянути та розв’язати задачі комбінаторної оптимізації з дробово-лінійною цільовою функцією на загальній множині розміщень.

Магістерська робота є продовженням і розвитком досліджень властивостей математичних моделей, у рамках теорії евклідової комбінаторної оптимізації на множині розміщень, задач з дробово-лінійною цільовою функцією, що є необхідним для пошуку розв’язків таких задач.

Метою роботи є побудова методів та алгоритмів розв'язування евклідових задач комбінаторної оптимізації на розміщеннях з дробово-лінійною функцією цілі на основі властивостей області допустимих розв'язків таких задач.

Дана мета конкретизується в таких задачах:

1. Одержання в аналітичному вигляді розв'язку безумовної задачі на розміщеннях.
2. Розробка методу та алгоритму розв'язування умовних задач на загальній множині розміщень; дослідження їх ефективності шляхом проведення числових експериментів.

Об'єкт дослідження – комбінаторні задачі оптимізації та методи їх розв'язування.

Предмет дослідження – евклідові задачі комбінаторної оптимізації з дробово-лінійною цільовою функцією на розміщеннях, методи їх розв'язування.

Методи дослідження. Основними методами, що використовувалися для проведення досліджень в роботі, є методи евклідової комбінаторної оптимізації [2], а також методи математичного програмування (для одержання розв'язків задач оптимізації з дробово-лінійною функцією цілі на загальній множині розміщень).

Наукова новизна одержаних результатів.

1. Аналітично розв'язана безумовна задача оптимізації на розміщеннях з дробово-лінійною цільовою функцією, що дає можливість, зокрема, використовувати це при розв'язуванні більш складних (умовних) задач на розміщеннях.
2. Запропоновано метод розв'язування умовних задач з дробово-лінійною функцією цілі на загальній множині розміщень шляхом розвинення методу розв'язування лінійних умовних задач на розміщеннях.

Теоретичне значення роботи. Теоретичні результати роботи можуть бути використані при подальшому дослідженні евклідових комбінаторних задач

оптимізації з дробово-лінійними функціями цілі на інших комбінаторних та полікомбінаторних множинах.

Практичне значення отриманих результатів. Розв'язок умовної оптимізаційної задачі з дробово-лінійною цільовою функцією на загальній множині розміщень дозволяє вирішувати конкретні задачі, які виникають з практичних потреб, коли на змінні накладаються обмеження не лише комбінаторного характеру, а ще деякі додаткові обмеження, зокрема, лінійні.

Кваліфікаційна робота складається з чотирьох розділів. У першому розділі описано постановку задачі. У другому розділі зроблено огляд відомостей з теорії евклідової комбінаторної оптимізації та оптимізаційних задач з дробово-лінійною цільовою функцією. У третьому розділі запропоновано метод розв'язування безумовної задачі з дробово-лінійною цільовою функцією на загальній множині розміщень. Четвертий розділ присвячено умовній задачі оптимізації та побудові методу лексикографічної еквівалентності. У додатках представлено блок-схеми алгоритмів розв'язування та приклади з серії числових експериментів.

Обсяг пояснювальної записки до кваліфікаційної роботи складає: 92 сторінки, в тому числі з яких: основна частина – 76 сторінок та 16 сторінок додатку, літературних джерел – 26 назв.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ

Задачі оптимізації з дробово-лінійною цільовою функцією знаходять широке застосування в різних галузях, таких як економіка, логістика, телекомунікації та управління ресурсами. Врахування дробової структури функції дозволяє моделювати реальні процеси, де ефективність характеризується співвідношенням двох лінійних величин (наприклад, прибутку до витрат, продуктивності до часу). Задачі такого типу є складними через нелінійний характер цільової функції та дискретну природу простору допустимих рішень, зокрема на множині розміщень. Це обґрунтовує необхідність розробки ефективних методів їх розв'язання.

Таким чином, необхідно:

1. Провести аналіз існуючих підходів до розв'язання задач оптимізації з дробово-лінійною цільовою функцією.
2. Сформулювати математичну модель задачі оптимізації на множині розміщень із врахуванням обмежень і специфіки простору розв'язків.
3. Розробити ефективний метод (або модифікувати існуючі методи) для розв'язання задачі з дробово-лінійною цільовою функцією.
4. Виконати теоретичний аналіз збіжності та ефективності запропонованого методу.
5. Реалізувати алгоритм у вигляді програмного забезпечення та провести експериментальне моделювання на тестових і прикладних задачах.
6. Проаналізувати отримані результати та запропонувати рекомендації щодо практичного застосування розробленого підходу.

РОЗДІЛ 2

ОСНОВНІ ВІДОМОСТІ З ТЕОРІЇ ЕВКЛІДОВОЇ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ ТА ОПТИМІЗАЦІЙНІ ЗАДАЧІ З ДРОБОВО-ЛІНІЙНОЮ ЦІЛЬОВОЮ ФУНКЦІЄЮ

2.1. Розвиток теорії комбінаторної оптимізації

Евклідова комбінаторна оптимізація виникла як математичне формулювання широкого спектра задач у галузі геометричного проектування. Вона охоплює комбінаторні оптимізаційні задачі, в яких допустимі конфігурації можуть бути представлені у вигляді точок в евклідовому просторі. Основи цієї теорії були розроблені членом-кореспондентом НАН України, доктором технічних наук, професором Ю. Г. Стояном.

Подальшою розробкою теорії і методів евклідової комбінаторної оптимізації разом зі Ю. Г. Стояном займалися С. В. Яковлев, О. О. Ємець та їх школи [1-18; 22-26].

Як показує аналіз публікацій, розвиток теорії евклідової комбінаторної оптимізації відбувається у двох напрямках:

- дослідження математичних моделей задач оптимізації на комбінаторних множинах.
- розробка та обґрунтування методів розв'язування оптимізаційних задач на комбінаторних множинах.

У ряді робіт (наприклад, [2; 4; 7; 18; 22-26]) досліджено побудову математичних моделей задач оптимізації на евклідових комбінаторних множинах, зокрема множинах сполучень, перестановок, поліперестановок, розміщень і полірозміщень. У роботах [3; 13] представлено моделі задач евклідової комбінаторної оптимізації з дробово-лінійною функцією цілі, визначених на множині перестановок.

Задачі з дробово-лінійною функцією цілі без комбінаторних обмежень та методи їх розв'язання розглядалися раніше (див., наприклад, [19-21]). Проте запропоновані моделі не завжди повністю відображають властивості задач або їхню специфіку. У випадках, коли допустимі розв'язки мають властивості, такі як належність до множини розміщень, доцільно формулювати задачу як комбінаторну оптимізацію на множині розміщень. Такі задачі з дробово-лінійною цільовою функцією ще не досліджені і потребують подальшого аналізу.

Задача з дробово-лінійною цільовою функцією є оптимізацією відношення двох лінійних функцій. Виникнення таких задач на множинах розміщень створює потребу в розробці ефективних методів і алгоритмів, здатних розв'язувати як окремі задачі, так і цілі їх класи. Це вимагає детального вивчення властивостей задач, включаючи аналіз їх областей визначення.

Як показано в [2], опуклою оболонкою загальної множини розміщень є многогранник розміщень. При його аналітичному представленні многогранник описується як множина розв'язків системи лінійних нерівностей, кожна з яких визначає гіпергрань максимальної вимірності многогранника. Така форма подання особливо важлива для пошуку розв'язків задач лінійної дискретної оптимізації, оскільки дає змогу застосовувати методи лінійного програмування.

У теорії систем лінійних нерівностей важливим є поняття незвідної системи — такої, яка не містить надлишкових обмежень. Це має особливе значення для задач на комбінаторних множинах, оскільки зі зростанням вимірності системи кількість обмежень збільшується, що ускладнює обчислення.

Дослідження незвідних систем для опуклих оболонок комбінаторних множин, таких як перестановки і розміщення, а також для задач з дробово-лінійною функцією цілі висвітлені в роботах [3; 15; 13].

1.2. Огляд методів розв'язування комбінаторних задач

Виникнення оптимізаційних задач на евклідових комбінаторних множинах потребує створення ефективних методів і алгоритмів для їх вирішення. Як показує аналіз літератури (наприклад, [2; 3; 5; 6; 13-24]), до точних підходів відносять дві основні групи методів: методи відсікання та комбінаторні методи.

Методи цієї групи ґрунтуються на виключенні частин допустимої області шляхом побудови правильних відсікань. Концепція відсікання була запропонована Гоморі [20], який, спираючись на зв'язок між задачами цілочислового та лінійного програмування, встановлений Дж. Данцигом [19], розробив алгоритми для цілочислових задач. Суть методу полягає в послідовному виключенні частин релаксованої області (без вимоги цілочисельності змінних), які не містять оптимальних рішень.

Ефективність цього підходу залежить від способу побудови відсікань, що породило численні варіації методу. Методи відсікання також адаптовані для евклідової комбінаторної оптимізації, зокрема для задач на вершинно-розташованих множинах. У таких задачах відсікають вершини комбінаторного многогранника, замінюючи умову належності до комбінаторної множини умовою належності до її опуклої оболонки.

Комбінаторні методи спираються на скінченну кількість допустимих рішень. Їх ідея полягає в частковому переборі "перспективних" варіантів шляхом відкидання завідомо неоптимальних. Найбільш відомі представники цієї групи – метод гілок і меж, метод динамічного програмування, а також наближені локальні методи.

Для евклідової комбінаторної оптимізації розроблено метод напрямленого перебору з використанням лексикографічного порядку, що дозволяє структурувати пошук розв'язків. Зокрема, цей підхід реалізований у методі лексикографічної еквівалентності для задач із лінійною цільовою функцією [16].

Для задач великої розмірності використовують метод послідовного приєднання обмежень (МППО) [2], який ідеологічно нагадує метод відсікання гіперплощин. Він уточнює допустиму область шляхом додавання лише активних обмежень, але може призводити до накопичення надлишкових обмежень, що ускладнює обчислення. Для оптимізації обчислювальних ресурсів застосовують процедуру відкидання несуттєвих обмежень [21].

Методи відсікання мають недолік у чутливості до обчислювальних похибок, які можуть призводити до не оптимальних рішень. Натомість комбінаторні методи демонструють простоту реалізації завдяки скінченності множини розв'язків.

Таким чином, розробка і вдосконалення методів для евклідових комбінаторних задач залишається важливим напрямом досліджень, особливо з огляду на їх різноманітні застосування.

РОЗДІЛ 3

МЕТОД РОЗВ'ЯЗУВАННЯ БЕЗУМОВНОЇ ЗАДАЧІ З ДРОБОВО-ЛІНІЙНОЮ ФУНКЦІЄЮ ЦІЛІ НА РОЗМІЩЕННЯХ

У даному розділі формулюється постановка безумовної задачі оптимізації з дробово-лінійною функцією цілі на загальній множині розміщень, доводиться критерій одержання її оптимального розв'язку.

3.1. Постановка задачі

Нехай k, n, η – натуральні константи, g_j, e_i – додатні дійсні числа $\forall j \in J_\eta, \forall i \in J_n$, а $G = \{g_1, g_2, \dots, g_\eta\}$ – мультимножина з основою $S(G) = (e_1, e_2, \dots, e_n)$ та кратностями елементів $k_G(e_i) = \eta_i, i \in J_n$. Загальну множину k -розміщень позначимо $E_{\eta n}^k(G)$.

Нехай також $x = (x_1, x_2, \dots, x_k) \in R^k$; $c_0, d_0 \in R$; $c_i, d_i \in R \quad \forall i \in J_k$.

Розглянемо евклідову задачу комбінаторної оптимізації з дробово-лінійною функцією цілі на загальній множині розміщень: знайти впорядковану пару $\langle F(x^*), x^* \rangle$, таку що

$$F(x^*) = \underset{x \in R^k}{extr} \frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0}, \quad (3.1)$$

$$x^* = \arg \underset{x \in R^k}{extr} \frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0}, \quad (3.2)$$

за комбінаторної умови

$$x = (x_1, \dots, x_k) \in E_{\eta^n}^k(G) \subset R^k. \quad (3.3)$$

Тут і далі вважаємо, що $\sum_{i=1}^k d_i x_i + d_0 > 0 \quad \forall x \in E_{\eta^n}^k(G)$.

Для дослідження задачі (3.1)-(3.3) перейдемо до задачі з лінійною функцією цілі. Позначимо:

$$y_0 = \frac{1}{\sum_{i=1}^k d_i x_i + d_0}, \quad y_i = x_i y_0 \quad \forall i \in J_k, \quad x \in R^k. \quad (3.4)$$

Співвідношення (3.4), зокрема, задають відображення $\psi(E_{\eta^n}^k(G)) = E \subset R^{k+1}$, причому $y_0 > 0, \quad y_i \geq 0 \quad \forall i \in J_k, \quad \psi(x) = y = (y_0, y_1, \dots, y_k)$.

За цих умов задача (3.1)-(3.3) зводиться до знаходження упорядкованої пари $\langle F'(y^*), y^* \rangle$

$$F'(y^*) = F(x^*) = \underset{y \in R^{k+1}}{extr} \sum_{i=0}^k c_i y_i, \quad (3.5)$$

$$y^* = \arg \underset{y \in R^{k+1}}{extr} \sum_{i=0}^k c_i y_i, \quad (3.6)$$

за умови

$$y = (y_0, y_1, \dots, y_k) \in E \subset R^{k+1} \quad (3.7)$$

Покажемо що, якщо $y^*=(y_0^*, y_1^*, \dots, y_k^*)$ – розв'язок задачі (3.5)-(3.7) з лінійною функцією цілі, то $x^*=(x_1^*, \dots, x_k^*)$ – розв'язок задачі (3.1)-(3.3).

Виберемо довільний допустимий розв'язок $y=(y_0, y_1, \dots, y_k)$ задачі з лінійною функцією цілі. Оскільки $y^*=(y_0^*, y_1^*, \dots, y_k^*)$ задовольняє умову (3.6), має місце наступна нерівність $\sum_{i=0}^k c_i y_i \leq \sum_{i=0}^k c_i y_i^*$, якщо в (3.5), (3.6) функція максимізується, $(\sum_{i=0}^k c_i y_i \geq \sum_{i=0}^k c_i y_i^*,$ якщо цільова функція мінімізується).

Враховуючи співвідношення (3.4), дані нерівності перепишуться так:

$$\frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0} \leq y_0^* \left(\sum_{i=1}^k c_i x_i^* + c_0 \right), \left(\frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0} \geq y_0^* \left(\sum_{i=1}^k c_i x_i^* + c_0 \right) \right).$$

Поділимо праві частини нерівностей на $\sum_{i=1}^k d_i y_i^* + d_0 y_0^* = 1$, і скоротимо їх на y_0^* , отримаємо

$$\frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0} \leq \frac{\sum_{i=1}^k c_i x_i^* + c_0}{\sum_{i=1}^k d_i x_i^* + d_0}, \left(\frac{\sum_{i=1}^k c_i x_i + c_0}{\sum_{i=1}^k d_i x_i + d_0} \geq \frac{\sum_{i=1}^k c_i x_i^* + c_0}{\sum_{i=1}^k d_i x_i^* + d_0} \right).$$

З останніх нерівностей переконуємося в оптимальності розв'язку $x^*=(x_1^*, \dots, x_k^*)$ для вихідної задачі (3.1)-(3.3) у випадку максимізації (мінімізації) цільової функції.

Якщо $\sum_{i=1}^k d_i x_i + d_0 < 0$, то одержимо задачу протилежну за знаком до задачі (3.1)-(3.3).

Отже, комбінаторна задача з дробово-лінійною функцією цілі на загальній множині розміщень може бути зведена до задачі з лінійною функцією цілі, при цьому кількість змінних збільшиться на одну, а до системи обмежень додається рівняння $\sum_{i=1}^k d_i y_i + d_0 y_0 = 1$.

Для розв'язування задачі (3.5)-(3.7) важливо знати структуру опуклої оболонки множини E . Опуклою оболонкою загальної множини розміщень $E_{\eta^n}^k(G)$, як відомо, [2], є загальний многогранник розміщень $\Pi_{\eta^n}^k(G)$. Важливим також для подальшого дослідження є той факт, що елементи множини $E_{\eta^n}^k(G)$ містяться всередині многогранника $\Pi_{\eta^n}^k(G)$.

Загальний многогранник розміщень в аналітичному вигляді, як відомо [2], задається наступною системою лінійних нерівностей:

$$\begin{cases} \sum_{j=1}^{|\omega|} g_j \leq \sum_{i \in \omega} x_i & \forall \omega \subset J_k, \\ \sum_{i \in \omega} x_i \leq \sum_{j=1}^{|\omega|} g_{\eta-j+1} & \forall \omega \subset J_k. \end{cases} \quad (3.8)$$

Застосуємо перетворення (3.4) до точок, що описуються системою (3.8), отримаємо

$$\sum_{j=1}^{|\omega|} g_j y_0 \leq \sum_{i \in \omega} y_i \quad \forall \omega \subset J_k, \quad (3.9)$$

$$\sum_{i \in \omega} y_i \leq \sum_{j=1}^{|\omega|} g_{\eta-j+1} y_0 \quad \forall \omega \subset J_k, \quad (3.10)$$

$$\sum_{i=1}^k d_i y_i + d_0 y_0 = 1, \quad (3.11)$$

$$y_0 > 0, \quad y_i \geq 0 \quad \forall i \in J_k. \quad (3.12)$$

Множину, яка задана системою (3.9)-(3.12), позначимо $Q_{\eta_n}^k(G) \in R^{k+1}$. Система, що складається з обмежень (3.9), (3.10), визначає опуклий многогранний конус із вершиною в точці $O(0,0,...,0)$, а (3.11) – гіперплощину, яка перерізає конус, не проходячи через його вершину. Тоді многогранник $Q_{\eta_n}^k(G)$ – основа піраміди, утвореної конусом та гіперплощиною, що лежить у частині R^{k+1} , у якій усі координати точок додатні.

3.2. Розв'язування безумовної задачі з дробово-лінійною цільовою функцією на множині розміщень

Побудуємо метод розв'язування безумовної евклідової комбінаторної задачі на розміщеннях з дробово-лінійною цільовою функцією вигляду (3.1)-(3.3) для випадку, коли цільова функція мінімізується. Після перетворення (3.4) задача (3.1)-(3.3) зведеться до знаходження пари $\langle F'(y^*), y^* \rangle$:

$$F'(y^*) = \min_{x \in R^{k+1}} \sum_{i=0}^k c_i y_i, \quad y^* = \arg \min_{x \in R^{k+1}} \sum_{i=0}^k c_i y_i, \quad (3.13)$$

за комбінаторної умови

$$y = (y_0, y_1, ..., y_k) \in E \subset R^{k+1} \quad (3.14)$$

та додатковому обмеженні

$$\sum_{i=1}^k d_i y_i + d_0 y_0 = 1. \quad (3.15)$$

Як відомо, $\text{conv } E = Q_{\eta n}^k(G)$, де многогранник $Q_{\eta n}^k(G)$ лежить в гіперплощині (3.15) простору R^{k+1} .

У лемі 3.3 доведено, що при відображенні (3.4) суміжні вершини многогранника розміщень $\Pi_{\eta n}^k(G)$ переходять у суміжні вершини $Q_{\eta n}^k(G)$.

Як наслідок теореми про суміжність вершин многогранника $\Pi_{\eta n}^k(G)$, справедливим є наступний критерій суміжності вершин многогранника $Q_{\eta n}^k(G)$, в якому вершину многогранника $Q_{\eta n}^k(G)$ зручно представити у вигляді

$$y^* = (y_0, y_1, \dots, y_k) = \frac{1}{D_\alpha} (1, g_{\alpha_1}, \dots, g_{\alpha_k}) = \frac{1}{D_\alpha} (1, g(\alpha)), \quad \text{де} \quad D_\alpha = \sum_{t=1}^k d_t g_{\alpha_t} + d_0, \\ g_{\alpha_t} \in \{g_1, g_2, \dots, g_s, g_{\eta-r+1}, \dots, g_\eta\} \quad \forall t \in J_k, \quad r, s \in J_k^0, \quad s+r=k.$$

Теорема 3.1. (Критерій суміжності вершин многогранника $Q_{\eta n}^k(G)$).

Вершинами многогранника $Q_{\eta n}^k(G)$, суміжними з вершиною

$$y^* = (y_0, y_1, \dots, y_k) = \frac{1}{D_\alpha} (1, g_{\alpha_1}, \dots, g_{\alpha_k}), \quad \text{де} \quad g_{\alpha_t} \in \{g_1, g_2, \dots, g_s, g_{\eta-r+1}, \dots, g_\eta\}, \quad \forall t \in J_k,$$

$r, s \in J_k^0, \quad s+r=k, \quad D_\alpha = \sum_{t=1}^k d_t g_{\alpha_t} + d_0$, є усякі вершини, утворені з y^* або

переставленням у векторі $g(\alpha) = (g_{\alpha_1}, \dots, g_{\alpha_k})$ компонент рівних g_i, g_{i+1} , де

$g_i \neq g_{i+1}, \quad i \in J_{s-1} \cup (J_{\eta-1} \setminus J_{\eta-r})$, або заміною будь-якої компоненти вектора

$g(\alpha)$ точки y^* рівної g_s ($g_{\eta-r+1}$) на $g_{\eta-r}$ (g_{s+1}), де відповідно

$g_s \neq g_{\eta-r} (g_{\eta-r+1} \neq g_{s+1}), \quad r, \quad s \in J_k^0, \quad s+r=k, \quad \text{за відповідної заміни (або переставлення) у вершині } y^* = \frac{1}{D_\alpha}(1, g(\alpha)).$

Враховуючи відомий факт з теорії лінійного програмування [20], що у вершині y^* будь-якого многогранника мінімум лінійної цільової функції досягається, якщо і тільки якщо в суміжних вершинах y цього многогранника значення цільової функції $F'(y)$ не менше $F'(y^*)$, та використовуючи критерій суміжності вершин многогранника $Q_{\eta^n}^k(G)$ (теорема 3.1), розв'яжемо задачу (3.13)-(3.15).

Нехай $y^* = (y_0, y_1, \dots, y_k)$ – вершина многогранника $Q_{\eta^n}^k(G)$ в позначеннях теореми 3.1. Аналогічно позначимо y – суміжну вершину до даної y^* :

$$y = (y_0, y_1, \dots, y_k) = \frac{1}{D_\beta}(1, g_{\beta_1}, \dots, g_{\beta_k}) = \frac{1}{D_\beta}(1, g(\beta)), \quad \text{де}$$

$$g_{\beta_t} \in \{g_1, g_2, \dots, g_\delta, g_{\eta-\rho+1}, \dots, g_\eta\} \quad \forall t \in J_k, \quad \delta, \rho \in J_k^0, \quad \delta + \rho = k, \quad D_\beta = \sum_{t=1}^k d_t g_{\beta_t} + d_0.$$

Розглянемо всі можливі співвідношення s та δ ; r та ρ .

1. Вважаємо, що $s = \delta$, $r = \rho$, тоді суміжна вершина y до даної y^* утворена переставленням в $g(\alpha)$ компонент рівних g_i , g_{i+1} , де $g_i \neq g_{i+1}$, $i \in I = J_{s-1} \cup (J_{\eta-1} \setminus J_{\eta-r})$, де $r, s \in J_k^0$, $s+r=k$, тобто $g(\alpha)$ та $g(\beta)$ відрізняються парою компонент. Тоді, якщо в $g(\alpha)$:

$$g_{\alpha_m} = g_i, \quad g_{\alpha_q} = g_{i+1}, \quad i \in I, \quad m, q \in J_k, \quad (3.16)$$

то в $g(\beta)$ відповідні координати визначаються так:

$$g_{\beta_m} = g_{i+1}, \quad g_{\beta_q} = g_i, \quad i \in I, \quad m, q \in J_k. \quad (3.17)$$

Всі інші координати в y та y^* однакові, тобто $g_{\alpha_t} = g_{\beta_t} \quad \forall t \in J_k \setminus \{m, q\}$.

Якщо y^* задовольняє (3.13) за умов (3.14), (3.15), то, враховуючи (3.16), (3.17), має місце таке співвідношення:

$$\frac{C^* + c_m g_i + c_q g_{i+1}}{D^* + d_m g_i + d_q g_{i+1}} \leq \frac{C^* + c_m g_{i+1} + c_q g_i}{D^* + d_m g_{i+1} + d_q g_i}, \quad (3.18)$$

де $C^* = \sum_{t \in I_1} c_t g_{\alpha_t} + c_0$, $D^* = \sum_{t \in I_1} d_t g_{\alpha_t} + d_0$, $I_1 = J_k \setminus \{m, q\}$, $i \in I$.

Враховуючи, що знаменники (3.18) задовольняють умову $\sum_{i=1}^k d_i x_i + d_0 > 0$,

нерівність (3.18) перепишеться так:

$$\begin{aligned} & (C^* + c_m g_i + c_q g_{i+1})(D^* + d_m g_{i+1} + d_q g_i) \leq \\ & (C^* + c_m g_{i+1} + c_q g_i)(D^* + d_m g_i + d_q g_{i+1}). \end{aligned}$$

Виконаємо перетворення останньої нерівності:

$$\begin{aligned} & C^* D^* + D^* (c_m g_i + c_q g_{i+1}) + C^* (d_m g_{i+1} + d_q g_i) + \\ & + c_m d_q g_i^2 + c_m d_m g_i g_{i+1} + c_q d_q g_i g_{i+1} + c_q d_m g_{i+1}^2 \leq C^* D^* + D^* (c_q g_i + c_m g_{i+1}) + \\ & + C^* (d_q g_{i+1} + d_m g_i) + c_q d_m g_i^2 + c_m d_m g_i g_{i+1} + c_q d_q g_i g_{i+1} + c_m d_q g_{i+1}^2, \end{aligned}$$

отримаємо (з урахуванням того, що $g_i \neq g_{i+1}$) таке співвідношення:

$$D^* (c_q - c_m) + C^* (d_m - d_q) \leq (g_{i+1} + g_i) (c_m d_q - c_q d_m). \quad (3.19)$$

2. У випадку, коли $s \neq \delta$, $r \neq \rho$, суміжна вершина y до даної y^* утворена заміною будь-якої компоненти $g(\alpha)$ рівної g_s ($g_{\eta-r+1}$) на $g_{\eta-r}$ (g_{s+1}), де відповідно $g_s \neq g_{\eta-r}$ ($g_{\eta-r+1} \neq g_{s+1}$), $r, s \in J_k^0$, $s+r=k$. Можливі такі випадки:

2.1. Серед компонент $g(\alpha)$ є рівні g_s та $g_{\eta-r+1}$, тоді $g(\alpha)$ та $g(\beta)$ відрізняються однією компонентою. Якщо в $g(\alpha)$: $g_{\alpha_m} = g_s$, то в $g(\beta)$ відповідна координата рівна $g_{\beta_m} = g_{\eta-r}$ ($g_{\eta-r} \neq g_s$). Всі інші координати однакові, тобто $g_{\alpha_t} = g_{\beta_t} \quad \forall t \in J_k \setminus \{m\}$. Якщо y^* задовольняє (3.13) за умов (3.14), (3.15), то має місце таке співвідношення:

$$\frac{C^{**} + c_m g_s}{D^{**} + d_m g_s} \leq \frac{C^{**} + c_m g_{\eta-r}}{D^{**} + d_m g_{\eta-r}}, \quad (3.20)$$

де $C^{**} = \sum_{t \in I_2} c_t g_{\alpha_t} + c_0$, $D^{**} = \sum_{t \in I_2} d_t g_{\alpha_t} + d_0$, $I_2 = J_k \setminus \{m\}$.

Враховуючи, що знаменники (3.20) задовольняють умову $\sum_{i=1}^k d_i x_i + d_0 > 0$,

отримаємо після перетворення таку нерівність:

$$\begin{aligned} C^{**} D^{**} + D^{**} c_m g_s + C^{**} d_m g_{\eta-r} + c_m d_m g_s g_{\eta-r} &\leq \\ &\leq C^{**} D^{**} + D^{**} c_m g_{\eta-r} + C^{**} d_m g_s + c_m d_m g_s g_{\eta-r}, \end{aligned}$$

або в спрощеному вигляді: $D^{**} c_m (g_s - g_{\eta-r}) + C^{**} d_m (g_{\eta-r} - g_s) \leq 0$. Оскільки $g_{\eta-r} > g_s$, бо $s < \eta - r$, отримаємо

$$C^{**} d_m - D^{**} c_m \leq 0. \quad (3.21)$$

Якщо в $g(\alpha)$: $g_{\alpha_q} = g_{\eta-r+1}$, то в $g(\beta)$ відповідна координата рівна $g_{\beta_q} = g_{s+1}$ ($g_{s+1} \neq g_{\eta-r+1}$). Усі інші координати однакові, тобто $g_{\alpha_t} = g_{\beta_t}$ $\forall t \in J_k \setminus \{q\}$. Враховуючи, що y^* задовольняє (3.13) за умов (3.14), (3.15), аналогічно попереднім міркуванням отримаємо:

$$D^{***} c_q (g_{\eta-r+1} - g_{s+1}) + C^{***} d_q (g_{s+1} - g_{\eta-r+1}) \leq 0, \quad C^{***} = \sum_{t \in I_3} c_t g_{\alpha_t} + c_0,$$

$$D^{***} = \sum_{t \in I_3} d_t g_{\alpha_t} + d_0, \quad I_3 = J_k \setminus \{q\}. \quad \text{Оскільки } g_{\eta-r+1} > g_{s+1}, \text{ остання нерівність}$$

запишеться так:

$$D^{***} c_q - C^{***} d_q \leq 0. \quad (3.22)$$

2.2. Серед компонент $g(\alpha)$ точки y^* є рівні g_s , але немає рівних $g_{\eta-r+1}$, тоді має місце співвідношення (3.21).

2.3. Серед компонент $g(\alpha)$ точки y^* є рівні $g_{\eta-r+1}$, але немає рівних g_s , тоді має місце співвідношення (3.22).

Таким чином, доведена така теорема.

Теорема 3.7. Для того, щоб точка $y^* = (y_0, y_1, \dots, y_k) =$
 $= \frac{1}{D_\alpha} (1, g_{\alpha_1}, \dots, g_{\alpha_k}) = \frac{1}{D_\alpha} (1, g(\alpha))$, де $g_{\alpha_t} \in \{g_1, g_2, \dots, g_s, g_{\eta-r+1}, \dots, g_\eta\}$, $\forall t \in J_k$, r ,
 $s \in J_k^0$, $s + r = k$, $D_\alpha = \sum_{t=1}^k d_t g_{\alpha_t} + d_0$, була мінімаллю функції $F'(y) = \sum_{i=0}^k c_i y_i$ (за
комбінаторної умови $y = (y_0, y_1, \dots, y_k) \in E$, де $E = \psi(E_{\eta}^k(G))$, а $y_0 > 0$ та
 $\sum_{i=0}^k d_i y_i = 1$) необхідно і достатньо, щоб для будь-яких компонент $g_{\alpha_m} = g_i$,
 $g_{\alpha_q} = g_{i+1}$ вектора $g(\alpha)$ таких, що $i \in J_{s-1} \cup (J_{\eta-1} \setminus J_{\eta-r})$, де $r, s \in J_k^0$, $s + r = k$,
виконувалась умова

$$D^*(c_q - c_m) + C^*(d_m - d_q) \leq (g_{i+1} + g_i)(c_m d_q - c_q d_m),$$

а для будь-яких компонент g_{α_m} , g_{α_q} вектора $g(\alpha)$ точки y^* рівних $g_{\alpha_m} = g_s$ і $g_{\alpha_q} = g_{\eta-r+1}$ виконувались відповідно умови

$$C^{**} d_m - D^{**} c_m \leq 0 \text{ і } D^{***} c_q - C^{***} d_q \leq 0,$$

де

$$C^* = \sum_{t \in I_1} c_t g_{\alpha_t} + c_0, \quad D^* = \sum_{t \in I_1} d_t g_{\alpha_t} + d_0, \quad I_1 = J_k \setminus \{m, q\}; \quad (3.23)$$

$$C^{**} = \sum_{t \in I_2} c_t g_{\alpha_t} + c_0, \quad D^{**} = \sum_{t \in I_2} d_t g_{\alpha_t} + d_0, \quad I_2 = J_k \setminus \{m\}; \quad (3.24)$$

$$C^{***} = \sum_{t \in I_3} c_t g_{\alpha_t} + c_0, \quad D^{***} = \sum_{t \in I_3} d_t g_{\alpha_t} + d_0, \quad I_3 = J_k \setminus \{q\}. \quad (3.25)$$

Зауваження. Враховуючи, що задачі (3.1)-(3.3) та (3.13)-(3.15) еквівалентні (тобто, якщо $y^* = (y_0, y_1, \dots, y_k)$ – розв’язок задачі (3.13)-(3.15), то розв’язок $x^* = (x_1, x_2, \dots, x_k)$ задачі (3.1)-(3.3) отримаємо із співвідношення $x_i = \frac{y_i}{y_0} \quad \forall i \in J_k$,

що випливає з перетворення (3.4)), аналогічна теорема має місце і для точки x^* .

Теорема 3.3. (Критерій оптимальності безумовної задачі з дробово-лінійною цільовою функцією на розміщеннях). Для того, щоб точка x^* була

мінімаллю функції
$$F(x) = \frac{\sum_{j=1}^k c_j x_j + c_0}{\sum_{j=1}^k d_j x_j + d_0} \quad (\text{за комбінаторної умови})$$

$x = (x_1, \dots, x_k) \in E_{\eta}^k(G)$ та з вимоги $\sum_{i=1}^k d_i x_i + d_0 > 0$) необхідно і достатньо, щоб для будь-яких координат $g_{\alpha_m} = g_i$, $g_{\alpha_q} = g_{i+1}$, точки x^* , таких, що $i \in J_{s-1} \cup (J_{\eta-1} \setminus J_{\eta-r})$, де $r, s \in J_k^0$, $s+r=k$, виконувалась умова $D^*(c_q - c_m) + C^*(d_m - d_q) \leq (g_{i+1} + g_i)(c_m d_q - c_q d_m)$, а для будь-яких координат точки x^* рівних g_s і $g_{\eta-r+1}$ виконувались відповідно умови $C^{**} d_m - D^{**} c_m \leq 0$ та $D^{***} c_q - C^{***} d_q \leq 0$, де $C^*, D^*; C^{**}, D^{**}; C^{***}, D^{***}$ визначаються відповідно співвідношеннями (3.23), (3.24), (3.25).

Не виконання однієї з умов теореми 3.3 свідчить про те, що вершина x , суміжна до x^* , надає цільовій функції меншого значення, ніж вершина x^* . Якщо така вершина одна, переходимо до цієї вершини та перевіряємо для неї умови теореми 3.3. Якщо декілька суміжних вершин $x \in X$, де $X = \text{vert } \Pi_{\eta}^k(G)$ – множина вершин многогранника розміщень, не задовольняють критерій оптимальності, вибираємо з цих вершин ту, для якої виконується умова:

$$\max_{x \in X} |F(x) - F(x^*)|. \quad (3.26)$$

З умови (3.26) шляхом перетворень, аналогічних доведенню теореми 3.2, отримаємо, що переходити треба до тієї з суміжних вершин, для якої спостерігається найбільше порушення однієї з умов (3.19), (3.21), (3.22) теореми 3.3, тобто:

$$\Delta_1 = \max_{x \in X} (D^*(c_q - c_m) + C^*(d_m - d_q) - (g_{i+1} + g_i)(c_m d_q - c_q d_m)),$$

$$\Delta_2 = \max_{x \in X} (C^{**} d_m - D^{**} c_m),$$

$$\Delta_3 = \max_{x \in X} (D^{***} c_q - C^{***} d_q).$$

Отже, для розв'язування задачі (3.1)-(3.3) для довільно вибраної вершини x многогранника розміщень $\Pi_{\eta}^k(G)$ перевіряємо умови теореми 3.3. Якщо x не задовольняє одну із умов (3.19), (3.21) чи (3.22), переходимо до наступної (суміжної) вершини з меншим значенням цільової функції до тих пір, поки не виконається критерій оптимальності.

Розглянемо приклад. Знайти мінімум функції $F(x) = \frac{-x_1 - x_2 - x_3 - x_4}{2x_1 + 5x_2 + 2x_3 + 4x_4}$

за умови $x = (x_1, \dots, x_4) \in E_{55}^4(G)$, де $G = \{2, 3, 4, 5, 7\}$.

Розв'язування.

$$x^1 = (2, 3, 4, 5), \quad s = 4, \quad r = 0.$$

Перевіряємо умову (3.19). Якщо $i=1$, то $g_1=2$, $g_2=3$, $c_m=-1$, $d_m=2$, $c_q=-1$, $d_q=5$. $D^* = 2 \cdot 4 + 5 \cdot 4 = 28$, $C^* = -4 - 5 = -9$. Підставимо усі значення в (3.19), отримаємо: $27 \leq -15$. Отже, умова (3.19) не виконується для $i=1$. Обчислимо порушення $\Delta_1(x^1) = 27 + 15 = 42$.

Якщо $i=2$, то $g_2=3$, $g_3=4$, $c_m=-1$, $d_m=5$, $c_q=-1$, $d_q=2$. $D^* = 24$, $C^* = -7$. Після підстановки в (3.19), отримаємо: $-21 \leq 21$. Отже, умова (3.19) виконується для $i=2$.

Якщо $i=3$, то $g_3=4$, $g_4=5$, $c_m=-1$, $d_m=2$, $c_q=-1$, $d_q=4$. $D^* = 19$, $C^* = -5$. Після підстановки в (3.19), отримаємо: $10 \leq -18$. Отже, умова (3.19) не виконується для $i=3$. Обчислимо порушення $\Delta_1(x^3) = 10 + 18 = 28$.

Перевіряємо умову (3.21), оскільки в x^1 є рівні $g_s = g_4 = 5$. З цільової функції $c_m = -1$, $d_m = 4$, $D^{**} = 27$, $C^{**} = -9$. Підставимо усі значення в (3.21), отримаємо: $-9 \leq 0$. Отже, умова (3.21) виконується.

Умову (3.22) не перевіряємо, оскільки серед координат x^1 немає рівних $g_{\eta-r+1}$. Враховуючи що $\Delta_1(x^1) = 42 > \Delta_1(x^3) = 28$, переходимо до перевірки вершини $x^2 = (3, 2, 4, 5)$, оскільки розглянута вершина не задовольняє умови теореми 3.3.

$$x^2 = (3, 2, 4, 5), s = 4, r = 0.$$

Для даної вершини не виконується умова (3.19), якщо $i = 3$. Переходимо до вершини $x^3 = (3, 2, 5, 4)$ і т.д., поки не отримаємо розв'язок $x^* = (5, 2, 7, 3)$. Оскільки вершина $x^* = (5, 2, 7, 3)$ задовольняє умови теореми 3.3, то $x^* = (5, 2, 7, 3)$ – мінімаль $F(x)$ на множині розміщень $E_{55}^4(G)$. Мінімальне значення функції $F(x^*) = -\frac{17}{46}$.

Таким чином, одержано критерій оптимальності безумовної задачі з дробово-лінійною цільовою функцією на загальній множині розміщень. Він, разом з критерієм суміжності вершин та критерієм вершини загального многогранника розміщень, дозволяє переходити від довільно вибраної початкової вершини до суміжної, аж до отримання екстремалі в розглянутій задачі.

РОЗДІЛ 4

МЕТОД РОЗВ'ЯЗУВАННЯ УМОВНИХ ЗАДАЧ З ДРОБОВО-ЛІНІЙНОЮ ЦІЛЬОВОЮ ФУНКЦІЄЮ НА ЗАГАЛЬНІЙ МНОЖИНІ РОЗМІЩЕНЬ

У даному розділі розглядається метод розв'язування задачі з дробово-лінійною функцією цілі та додатковими лінійними обмеженнями на розміщеннях. Для перевірки ефективності роботи алгоритму, побудованого на основі методу, проводиться аналіз числових експериментів.

4.1. Постановка задачі

Модель вигляду (3.1)-(3.3) не узагальнює властивості різних задач комбінаторної оптимізації на розміщеннях з дробово-лінійною функцією цілі, які виникають з практичних потреб. Досить часто на змінні накладаються обмеження не лише комбінаторного характеру, а ще деякі додаткові обмеження, зокрема, лінійні. Такі моделі матимуть інші властивості, та й пошук розв'язку їх дещо складніший. Тому доцільно розглянути умовні оптимізаційні задачі з дробово-лінійними цільовими функціями на загальній множині розміщень та методи їх розв'язування.

Нехай k, n, η, m, p – натуральні константи, $c_j, d_j \in R \quad \forall j \in J_m^0$, $a_{ij}, b_i \in R \quad \forall j \in J_m, \forall i \in J_p$, також $t = (t_1, t_2, \dots, t_k, t_{k+1}, \dots, t_m) \in R^m$, де $x_j = t_j \quad \forall j \in J_k$, тобто змінні $t_{k+1}, \dots, t_m$ є неперервними, а $x_1, \dots, x_k$ – комбінаторними.

Розглянемо задачу вигляду: знайти впорядковану пару $\langle F(t^*), t^* \rangle$, таку що

$$F(t^*) = \underset{t \in R^m}{extr} \frac{\sum_{j=1}^m c_j t_j + c_0}{\sum_{j=1}^m d_j t_j + d_0}, \quad (4.1)$$

$$t^* = \arg \underset{t \in R^m}{extr} \frac{\sum_{j=1}^m c_j t_j + c_0}{\sum_{j=1}^m d_j t_j + d_0}, \quad (4.2)$$

за комбінаторної умови

$$x = (x_1, \dots, x_k) \in E_{\eta^n}^k(G) \subset R^m, \quad k \leq m, \quad (4.3)$$

та додаткових лінійних обмеженнях

$$\sum_{j=1}^m a_{ij} t_j \leq b_i, \quad i \in J_p. \quad (4.4)$$

Вважаємо, що виконується умова $\sum_{j=1}^m d_j t_j + d_0 > 0$ для будь-якої точки $t = (t_1, t_2, \dots, t_m)$, що належить області визначення задачі (4.1)-(4.4) і цільова функція (4.1) максимізується. Якщо пара (4.1), (4.2) мінімізується, переходимо до задачі на максимум, змінивши коефіцієнти чисельника цільової функції на протилежні.

4.2. Метод побудови лексикографічної еквівалентності для задач з дробово-лінійною цільовою функцією на загальній множині розміщень

Задача (4.1)-(4.4) може бути розв'язана шляхом переходу до релаксованої задачі: умову (4.3) „послабимо”, замінивши її опуклою оболонкою множини $E_{\eta}^k(G)$:

$$\sum_{j=1}^{|\omega|} g_j \leq \sum_{i \in \omega} x_i \quad \forall \omega \subset J_k, \quad (4.5)$$

$$\sum_{i \in \omega} x_i \leq \sum_{j=1}^{|\omega|} g_{\eta-j+1} \quad \forall \omega \subset J_k. \quad (4.6)$$

Застосовуючи до задачі (4.1), (4.2), (4.4)-(4.6) відображення ψ , яке задамо співвідношеннями

$$y_0 = \frac{1}{\sum_{j=1}^m d_j t_j + d_0}, \quad z_j = t_j y_0 \quad \forall j \in J_m, \quad t \in R^m, \quad (4.7)$$

перейдемо до задачі з лінійною функцією цілі та додатковими лінійними обмеженнями, яка зводиться до знаходження упорядкованої пари $\langle F'(z^*), z^* \rangle$:

$$F'(z^*) = \max_{z \in R^{m+1}} \sum_{j=1}^m c_j z_j + c_0 y_0, \quad (4.8)$$

$$z^* = \arg \max_{z \in R^{m+1}} \sum_{j=1}^m c_j z_j + c_0 y_0 \quad (4.9)$$

за умов

$$\sum_{j=1}^m a_{ij} z_j - b_i y_0 \leq 0, \quad i \in J_p, \quad (4.10)$$

$$\sum_{j=1}^{|\omega|} g_j y_0 \leq \sum_{i \in \omega} y_i \quad \forall \omega \subset J_k, \quad (4.11)$$

$$\sum_{i \in \omega} y_i \leq \sum_{j=1}^{|\omega|} g_{\eta-j+1} y_0 \quad \forall \omega \subset J_k, \quad (4.12)$$

$$\sum_{j=1}^m d_j z_j + d_0 y_0 = 1, \quad y_0 > 0, \quad z_j \geq 0 \quad \forall j \in J_m, \quad (4.13)$$

де $z = (y_0, z_1, \dots, z_k, z_{k+1}, \dots, z_m) \in R^{m+1}$, $y_j = z_j \quad \forall j \in J_k$.

Аналогічно як і для задачі (3.5)-(3.7) можна показати, що якщо $z^* = (y_0^*, z_1^*, \dots, z_m^*)$, де $z_j = y_j \quad \forall j \in J_k$, – оптимальний розв’язок задачі (4.8)-(4.13),

то $t^* = (t_1^*, t_2^*, \dots, t_m^*)$, де $t_j^* = \frac{z_j^*}{y_0^*}$, $t_j = x_j \quad \forall j \in J_k$, є оптимальним розв’язком задачі

(4.1), (4.2), (4.4)-(4.6), тобто задачі (4.8)-(4.13) та (4.1), (4.2), (4.4)-(4.6) еквівалентні.

Таким чином, задача (4.1)-(4.4) є узагальненням умовної комбінаторної задачі з дробово-лінійною функцією цілі на множині розміщень.

У роботі [16] описано метод побудови лексикографічної еквівалентності для розв’язування задач евклідової комбінаторної оптимізації з лінійними функціями цілі та додатковими обмеженнями на загальній множині розміщень, в основі якого лежить розбиття простору за допомогою відношення еквівалентності та наступний напрямлений перегляд класів еквівалентності. Поширимо його для побудови методу розв’язування умовних задач з дробово-лінійною функцією цілі вигляду (4.1)-(4.4).

Розв'язування задачі (4.1)-(4.4) зводиться до розв'язування послідовності релаксованих задач лінійного програмування (4.8)-(4.13). Однак знайдений розв'язок $t^* = (t_1^*, t_2^*, \dots, t_m^*)$ задачі (4.1), (4.2), (4.4)-(4.6), де $t_j^* = \frac{z_j^*}{y_0^*}$, $\forall j \in J_k$, а $z^* = (y_0^*, z_1^*, \dots, z_m^*)$ – розв'язок задачі (4.8)-(4.13), не гарантує виконання умови (4.3) для вихідної задачі (4.1)-(4.4), оскільки накладаються додаткові умови (4.10) для розв'язуваної задачі. Враховуючи, що множина розміщень не збігається з множиною вершин своєї опуклої оболонки, тобто $E_{\eta^n}^k(G) \neq \text{vert conv } E_{\eta^n}^k(G)$, то згідно з теоремою 3.4. маємо, що множина E , що є образом $E_{\eta^n}^k(G)$ при перетворенні (4.7), теж не збігається з множиною вершин многогранника $Q_{\eta^n}^k(G)$. Тому пошук оптимального розв'язку, що є елементом множини $E_{\eta^n}^k(G)$, не може бути здійснений шляхом відсікання через вершини допустимої області допоміжної задачі, які є суміжними з вершиною, що відсікається [68]. Доцільним видається розглянути напрямлений перегляд точок множини $E_{\eta^n}^k(G)$, які задовольняють умову (4.3), аж до отримання максимуму та максималі задачі (4.1)-(4.4).

Далі пропонується метод розв'язування умовних задач з дробово-лінійною функцією цілі вигляду (4.1)-(4.4):

1. Здійснимо релаксацію задачі (4.1)-(4.4): умову (4.3) замінімо на (4.5)-(4.6).
2. До задачі (4.1), (4.2), (4.4)-(4.6) застосуємо перетворення (4.7).
3. Задачу (4.8)-(4.13) згідно з МППО [2] (τ – номер ітерації) запишемо у вигляді: знайти впорядковану пару $\langle F(z^\tau), z^\tau \rangle$, таку, що

$$F(z^\tau) = \max_{z \in D_\tau} \sum_{j=1}^m c_j z_j + c_0 y_0, \quad (4.14)$$

$$z^\tau = \arg \max_{z \in D_\tau} \sum_{j=1}^m c_j z_j + c_0 y_0 \quad (4.15)$$

на області D_τ , що описується системою S_τ , $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_m^\tau)$, де

$z_j^\tau = y_j^\tau \quad \forall j \in J_k$. При $\tau = 0$ система S_0 включає усі обмеження (4.10)-(4.13).

Для $\tau = 1$ формуємо систему S_1 лінійних обмежень, що описує область $D_1 \supset D_0$ і включає нерівності (4.10), (4.13) та (4.11), (4.12) при $|\omega| = 1$.

4. Розв'язуємо задачу (4.14)-(4.15) на області D_τ при значенні індексу $\tau (\tau \geq 1)$ модифікованим симплекс-методом. Якщо задача не має розв'язку, то не має розв'язку і (4.10)-(4.15), інакше нехай $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_m^\tau) \in R^{m+1}$ – розв'язок (4.10)-(4.15), отриманий на ітерації τ .

5. Якщо $\tau \leq 2$, то переходимо на пункт 6, інакше працює процедура відкидання обмежень (для зменшення вимірності розв'язуваної задачі) [140].

Нехай $f_{i,j} \leq 0$ – обмеження з номером j , під'єднане на ітерації i ($i = 2, 3, \dots, \tau - 1$), z^τ – точка, знайдена на ітерації τ , z^{i-1} – результат $(i-1)$ -ої ітерації, $F(z^\tau)$ – значення функції в точці z^τ . Якщо обмеження $f_{i,j} \leq 0$ ($i = 2, 3, \dots, \tau - 1$) задовольняє умови

$$f_{i,j}(z^\tau) < -\varepsilon, \quad (4.16)$$

$$F(z^\tau) \leq F(z^{i-1}) + f_{i,j}(z^{i-1}), \quad (4.17)$$

де $\varepsilon = 10^{-2} - 10^{-3}$ [38, 39], то дане обмеження виключається з системи S_τ .

6. За знайденою точкою $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_k^\tau, \dots, z_m^\tau) \in R^{m+1}$ визначаємо точку $y_0^\tau = (y_0^\tau, y_1^\tau, \dots, y_k^\tau)$ та перевіряємо виконання усіх обмежень системи (4.11),

(4.12). Якщо точка y^{τ} задовольняє не всі обмеження (4.11), (4.12), то переходимо на пункт 7, інакше – на пункт 8.

7. Приймаємо $\tau = \tau + 1$ та формуємо систему S_{τ} , включаючи до $S_{\tau-1}$ нерівності з (4.11), (4.12), що не виконуються в точці $y^{\tau-1}$. Відповідно S_{τ} описує область D_{τ} , $D_{\tau} \subset D_{\tau-1}$. Переходимо на пункт 4.

8. За розв'язком $y^{\tau} = (y_0^{\tau}, y_1^{\tau}, \dots, y_k^{\tau})$ задачі (4.10)-(4.15) визначаємо точку $\bar{x} = x^{\tau} = (x_1^{\tau}, x_2^{\tau}, \dots, x_k^{\tau})$ – розв'язок (4.1), (4.2), (4.4)-(4.6), де $x_i^{\tau} = \frac{y_i^{\tau}}{y_0^{\tau}} \forall i \in J_k$. Якщо при цьому $\bar{x} \in E_{\eta}^k(G)$, то задача (4.1)-(4.4) розв'язана, інакше шукаємо представників комбінаторних класів найближчого зліва та найближчого справа до класу з представником $\bar{x}$ (алгоритми див. нижче) [16]. Далі здійснюємо перегляд комбінаторних класів у порядку спадання і зростання, причому з розгляду виключаються точки, значення цільової функції яких менше, отриманого на попередніх ітераціях. Тоді одержана в результаті перебору точка дає розв'язок задачі (4.1)-(4.4).

Зауважимо, що для ефективної перевірки обмежень системи (4.11), (4.12) сформульована та доведена наступна теорема. Застосування цієї теореми дозволяє шукати розв'язок задачі (4.10)-(4.15) не на допустимій області, а на області, яка її містить. За рахунок цього кількість обмежень у системі значно зменшується і ефективніше працює МППО.

Теорема 4.1. Нехай $z^{\tau} = (y_0^{\tau}, z_1^{\tau}, \dots, z_k^{\tau}, \dots, z_m^{\tau}) \in R^{m+1}$, $y = (y_0, y_1, \dots, y_k) \in \text{conv } E$, де $E = \psi(E_{\eta}^k(G))$ [157, 164] та

$$y_i \leq y_{i+1} \quad \forall i \in J_{k-1}^0, \quad (4.18)$$

тоді з виконання для нерівності спілки за номером i обмежень

$$y_1 + y_2 + \dots + y_i \geq (g_1 + g_2 + \dots + g_i)y_0, \quad (4.19)$$

$$y_k + y_{k-1} + \dots + y_{k-i+1} \leq (g_{\eta} + g_{\eta-1} + \dots + g_{\eta-i+1})y_0 \quad (4.20)$$

впливає виконання всіх інших обмежень i -ої спільки в точці $y = (y_0, y_1, \dots, y_k)$.

Доведення. Враховуючи умови (4.18)-(4.20), для довільної нерівності i -ої спільки мають місце такі співвідношення:

$$y_{\alpha_1} + y_{\alpha_2} + \dots + y_{\alpha_i} \geq y_1 + y_2 + \dots + y_i \geq (g_1 + g_2 + \dots + g_i)y_0, \quad (4.21)$$

$$y_{\alpha_1} + y_{\alpha_2} + \dots + y_{\alpha_i} \leq y_k + y_{k-1} + \dots + y_{k-i+1} \leq (g_{\eta} + g_{\eta-1} + \dots + g_{\eta-i+1})y_0, \quad (4.22)$$

де $\alpha_j \in J_k \ \forall j \in J_i$. З (4.21), (4.22) випливає, що $\sum_{j=1}^{|\omega|} g_j y_0 \leq \sum_{i \in \omega} y_i \leq \sum_{j=1}^{|\omega|} g_{\eta-j+1} y_0$,

$\forall \omega \subset J_k$. А це і є система (4.11), (4.12). Теорема доведена.

Отже, для перевірки системи (4.11), (4.12) необхідно впорядкувати k перших координат точки y згідно з умовою (4.18) та перевірити по дві нерівності вигляду (4.19) та (4.20) з кожної спільки.

Таким чином, метод розв'язування задачі (4.1)-(4.4) містить у собі допоміжні задачі, що розв'язуються на певному кроці. Це знаходження оптимального розв'язку релаксованої задачі (4.1), (4.2), (4.4)-(4.6) за розв'язком лінійної задачі (4.10)-(4.15) шляхом перетворення (4.7); задачі пошуку комбінаторних класів найближчого зліва та справа до знайденого розв'язку $\bar{x} = x^{\tau} = (x_1^{\tau}, x_2^{\tau}, \dots, x_k^{\tau})$, що не задовольняє умову (4.3) вихідної задачі (4.1)-(4.4); задачі перебору комбінаторних класів у порядку зростання та спадання, починаючи з деякого комбінаторного класу. Алгоритми реалізації кожної із цих задач описані нижче. Їх розв'язування зводиться до задач лінійного програмування великої вимірності. Тому доцільним є використання методу

послідовного приєднання обмежень та відкидання на певній ітерації неактивних обмежень, алгоритми їх реалізації теж описані нижче.

4.3. Алгоритми, їх обґрунтування та оцінки

Алгоритм А методу розв'язування задачі (4.1)-(4.4) використовує в собі такі допоміжні алгоритми:

- алгоритм А1, що реалізує модифікований симплекс-метод для знаходження оптимального розв'язку релаксованої задачі (4.1), (4.2), (4.4)-(4.6) за розв'язком лінійної задачі (4.10)-(4.15) шляхом перетворення (4.7);
- алгоритм А2, який реалізує МППО для задачі (4.10)-(4.15);
- процедура відкидання неактивних обмежень оформлена у вигляді алгоритму А3;
- алгоритм А4 містить лексикографічну модифікацію двоїстого симплекс-методу;
- розв'язування задачі пошуку представника комбінаторного класу найближчого зліва до $\bar{V}$, представником якого є $\bar{x}$, описане в алгоритмі А5;
- алгоритм А6 реалізує пошук комбінаторного класу найближчого справа до $\bar{V}$, представником якого є $\bar{x}$;
- розв'язування задач перегляду комбінаторних класів у порядку спадання та зростання, починаючи з деякого комбінаторного класу, описані в алгоритмах А7 та А8 відповідно.

Розглянемо детальніше кожен із цих алгоритмів.

Алгоритм А1 модифікованого симплекс-методу. На вході в алгоритм задача (4.8)-(4.13) лінійного програмування, записана в канонічному вигляді, на виході – розв'язок у вигляді точки $z^T = (y_0^T, z_1^T, \dots, z_k^T, \dots, z_m^T) \in R^{m+1}$. Сам алгоритм та його блок-схема не викладені, оскільки він є загальновідомим [21].

Машинний час необхідний для розв'язування задачі визначається, головним чином, кількістю операцій множення та ділення, з якими зв'язаний

процес визначення оптимального плану. Операції порівняння, додавання та віднімання виконуються значно швидше, ніж множення та ділення.

По алгоритму А вперше алгоритм А1 застосовується до розв'язування задачі (4.14)-(4.15) на області D_1 , що описується системою S_1 . Проаналізуємо роботу алгоритму А1 для розв'язування цієї задачі.

На області D_1 простору R^{m+1} задача (4.14)-(4.15) описується системою $p+1+2C_k^1$ лінійних обмежень, де p – кількість додаткових лінійних обмежень (4.10), k – кількість комбінаторних змінних, $2C_k^1$ – кількість нерівностей першої спілки системи (4.11)-(4.12). У найгіршому випадку при зведенні до канонічного вигляду задача міститиме стільки базисних змінних, скільки всього обмежень, тобто $p+1+2C_k^1$, серед яких одна є штучною (внаслідок обмеження-рівності (4.13)). Тоді в задачі (4.14)-(4.15) стане $(m+1)+(p+1+2C_k^1)$ змінних. Для кожної окремої ітерації ділення виконується $2(p+1+2C_k^1)+1$ разів, а операція множення

$$\begin{aligned} & (p+1+2C_k^1)(2(p+1+2C_k^1)+2+(p+1+2C_k^1)+(m+1))= \\ & = (p+1+2k)(3(p+1+2k)+3+m) \end{aligned}$$

разів.

Зауваження. При визначенні вектору, що виводиться з базису, можлива неоднозначність, особливо для задач вигляду (4.8)-(4.13), які є виродженими. У цьому випадку працює процедура від зациклювання. У найгіршому випадку може бути виконано $(p+1+2C_k^1)+(p+2C_k^1)+(p+2C_k^1-1)+\dots+1$ ділень, або після спрощення $\frac{(p+1+2C_k^1)+1}{2}(p+1+2C_k^1)$. Тоді всього операцій ділення для

однієї ітерації алгоритму А1 буде виконано $\left(\frac{(p+1+2k)+1}{2}+1\right)(p+1+2k)+1$.

Оцінимо кількість операцій порівняння, додавання, віднімання, що виконуються при розв'язуванні задачі (4.14)-(4.15).

При визначенні компонент вектора, що вводиться в базис, та обчисленні оцінок основної таблиці, виконується по $(p+1+2C_k^1)(p+2C_k^1)$ операцій додавання, для підрахунку оцінок допоміжної таблиці – $(m+1)(p+2C_k^1)$. Всього операцій додавання для однієї ітерації алгоритму A1 буде виконано $(m+p+2+2k)(p+2k)$.

Для розрахунку оберненої матриці виконується $(p+1+2C_k^1)(p+2C_k^1+2)$ операцій віднімання і $(m+p+2+2C_k^1)-(p+1+2C_k^1)=m+1$ для підрахунку відносних оцінок допоміжної таблиці (обрахунки виконуються лише для небазисних змінних, оскільки для базисних відносні оцінки дорівнюють нулю). Разом одна ітерація алгоритму A1 містить $(p+1+2k)(p+2k+2)+m+1$ операцій віднімання.

При пошуку від'ємних відносних оцінок у найгіршому випадку переберемо $m+1$ елементів, для знаходження максимальної по модулю від'ємної відносної оцінки працює алгоритм сортування для $(m+1)$ елементів, при цьому виконується $(m+1)^2$ операцій. Для перевірки вектора, що вводиться в базис, на наявність додатних компонент, у найгіршому випадку виконується $p+1+2C_k^1$ операцій. Пошук мінімального відношення для виведення вектору з базису з урахуванням процедури зациклювання потребує $\frac{(p+1+2C_k^1)^3}{2}$ операцій порівняння. Всього операцій порівняння для однієї ітерації алгоритму A1 буде виконано

$$\frac{(p+1+2C_k^1)^3}{2} + (m+1)^2 + (m+1) + (p+1+2C_k^1).$$

Кількість ітерацій необхідних для розв'язування задачі (4.14), (4.15) залежить від початкового опорного розв'язку. Число ітерацій модифікованого симплекс-методу знаходиться в межах від $p+1+2k$ до $2(p+1+2k)$.

Алгоритм А2 методу послідовного приєднання обмежень. На вході в алгоритм задача лінійного програмування (4.14), (4.15) на області D_τ , що описується системою S_τ , на виході – розв'язок у вигляді точки $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_k^\tau, \dots, z_m^\tau) \in R^{m+1}$, система обмежень S_τ , що визначає область D_τ та лічильник кількості приєднаних обмежень q . На вході в алгоритм $q = 0$.

Крок 1. Розв'язуємо задачу лінійного програмування (4.14), (4.15) на області D_τ модифікованим симплекс-методом. Нехай $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_k^\tau, \dots, z_m^\tau) \in R^{m+1}$ – розв'язок на області D_τ .

Крок 2. За знайденою точкою $z^\tau = (y_0^\tau, z_1^\tau, \dots, z_k^\tau, \dots, z_m^\tau) \in R^{m+1}$ визначаємо $y^\tau = (y_0^\tau, y_1^\tau, \dots, y_k^\tau)$, де $y_j^\tau = z_j^\tau \quad \forall j \in J_k$. Впорядковуємо її координати по неспаданню згідно з умовою (4.18), отримаємо $y^0 = (y_0^\tau, y_{\alpha_1}^\tau, \dots, y_{\alpha_k}^\tau)$.

Крок 3. Перевіряємо по дві нерівності вигляду (4.19), (4.20) з кожної спілки в точці $y^0 = (y_0^\tau, y_{\alpha_1}^\tau, \dots, y_{\alpha_k}^\tau)$. Якщо точка задовольняє усі обмеження (4.11), (4.12) згідно з теоремою 4.1., то алгоритм закінчує свою роботу, інакше переходимо на наступний крок.

Крок 4. $\tau := \tau + 1$ і формуємо систему S_τ , включаючи до $S_{\tau-1}$ нерівності з систем (4.19), (4.20), що не виконуються у точці $y^{0^{-1}} = (y_0^{\tau-1}, y_{\alpha_1}^{\tau-1}, \dots, y_{\alpha_k}^{\tau-1})$. Відповідно S_τ описує область D_τ , $D_\tau \subset D_{\tau-1}$.

Крок 5. $q := q + |H_{\tau-1}|$, де $|H_{\tau-1}|$ – кількість приєднаних обмежень на ітерації $(\tau - 1)$, і переходимо на крок 1.

Дамо оцінку роботи алгоритму А2, використовуючи наведений алгоритм та його блок-схему (див. додаток А, рис. А.1). Як і в попередньому алгоритмі,

операції порівняння, додавання та віднімання виконуються значно швидше, ніж множення, ділення.

До формування системи S_1 , що описує область D_1 виконується 3 операції: 1 операція зчитування даних та 2 операції присвоєння. Далі реалізується алгоритм А1, який потребує, як описано вище, $(p+1+2C_k^1)(3(p+1+2C_k^1)+3+m)$ операцій множення та $\left(\frac{(p+1+2C_k^1)+1}{2}+1\right)(p+1+2C_k^1)+1$ ділень. Операцій додавання, віднімання та порівняння буде виконано всього для однієї ітерації симплекс-методу

$(m+p+2+2C_k^1)(p+2C_k^1)+(p+1+2C_k^1)(p+2C_k^1+2)+m+\frac{(p+1+2C_k^1)^3}{2}+1+(m+1)^2+(m+1)+(p+1+2C_k^1)$. Далі виконується 1 операція перевірки умови, чи має задача розв'язок. Для перевірки умови $\mathcal{U}^0 \in D_0$ необхідно:

- відсортувати за значенням по неспаданню компоненти точки $\mathcal{U}^0 = (y_0^\tau, y_{\alpha_1}^\tau, \dots, y_{\alpha_k}^\tau)$, для чого виконуємо в найгіршому випадку $(k+1)^2$ операцій порівняння;

- підставити компоненти відсортованої за значенням по неспаданню точки в нерівності спілки за номером i обмежень (4.19), (4.20) та перевірити правильність кожної з них для кожної $(k-1)$ спілки, тобто всього $2(k-1)$ нерівностей. Підрахуємо кількість операцій додавання. Легко помітити, що для перевірки нерівностей спілки $i=2$ необхідно 4 операції додавання, для $i=3$ – 8, для $i=4$ – 12, для $i=5$ – 16 і т.д. Утворена послідовність є арифметичною прогресією, сума $(k-1)$ -го членів якої дорівнює

$\frac{2 \cdot 4 + 4(k-1-1)}{2}(k-1) = 2k(k-1)$. Далі виконуємо $2(k-1)$ операцій множення (по дві в кожній $(k-1)$ спілці). З метою перевірки належності точки області D_0 виконуємо $2(k-1)$ операцій віднімання від правої частини ліву в нерівностях

(4.19) та від лівої частини праву в нерівностях (4.20), а далі проводимо $2(k-1)$ операцій порівняння отриманих результатів з нулем.

Кількість ітерацій МППО суттєво залежить від стратегії формування системи S_τ [22]. У запропонованій програмній реалізації алгоритму система S_τ утворюється шляхом залучення до $S_{\tau-1}$ по черзі для усіх спілок усіх нерівностей (4.19), (4.20), які не виконуються у точці $y_0^{\tau-1} = (y_0^{\tau-1}, y_{\alpha_1}^{\tau-1}, \dots, y_{\alpha_k}^{\tau-1})$. Максимальна кількість таких нерівностей дорівнює $2(k-1)$ (нерівності (4.19), (4.20) для кожної $(k-1)$ спілки). Кількість операцій, для однієї ітерації алгоритму A2 наведена в таблиці 4.1.

Таблиця 4.1

Кількість операцій однієї ітерації алгоритму A2

Операції	Кількість операцій
множення	$12k^2 + 3p^2 + 12kp + 2km + pm + 6p + m + 6k + 4$
ділення	$2k^2 + \frac{1}{2}p^2 + 2pk + 5k + p + 3$
додавання	$6k^2 + p^2 + mp + 2mk + 6kp + 2k + 2p$
віднімання	$4k^2 + p^2 + 4kp + 8k + 3p + m + 1$
порівняння	$: \frac{1}{2}p^3 + 4k^3$

Починаючи з другої ітерації, збільшується вимірність задачі, що розв'язується алгоритмом A1, у межах від 1 до $(k-1)$, що визначається кількістю приєднаних до системи S_τ обмежень. Враховуючи це, досить важко оцінити роботу алгоритму на наступних ітераціях. Однак, для алгоритму A2, без операцій алгоритму A1, для кожної ітерації кількість операцій є сталою, оскільки для перших $(k+1)$ координати точки y_0^k кожного разу перевіряється $2(k-1)$ нерівностей.

Оцінимо кількість ітерацій алгоритму A2, яка залежить від того, як швидко знайдеться точка, що буде задовольняти усі обмеження (4.10)-(4.13). Система (4.10)-(4.13) містить всього $p+1+2\sum_{i=1}^k C_k^i$ нерівностей, серед яких приєднано за алгоритмом A2 може бути не більше $2\sum_{i=2}^k C_k^i$, оскільки $p+1+2k$ нерівності визначають систему S_1 . Якщо вважати, що на кожній ітерації алгоритму A2 приєднується по одній нерівності вигляду (4.19) або (4.20), тому в найгіршому випадку ітерацій МППО буде не більше ніж $2^k - k - 1$. Однак, як показують результати числових експериментів, така кількість ітерацій практично ніколи не зустрічається. Кількість ітерацій МППО, як показує досвід, не перевищує $2m$.

Алгоритм A3, що реалізує процедуру відкидання неактивних обмежень. На вході в алгоритм – система S_τ , що описує область D_τ , на якій оптимізується цільова функція (4.14); z^τ – точка, знайдена на ітерації τ , z^{i-1} – результат $(i-1)$ -ої ітерації. На виході – модифікована система S_τ , що не містить неактивних обмежень та лічильник кількості відкинутих обмежень w . На вході в алгоритм $w=0$.

Крок 1. Перевіряємо для усіх обмежень $f_{i,j} \leq 0$ системи S_τ , які були приєднані за алгоритмом A2 до системи S_1 на ітерації i ($i=2,3,\dots,\tau-1$) і мають номер j , умову (4.16).

Крок 2. Якщо таких нерівностей немає, алгоритм A3 закінчує свою роботу, інакше формуємо множину W індексів обмежень, для яких умова $f_{i,j}(z^\tau) < -\varepsilon$ має місце і переходимо на крок 3.

Крок 3. Перевіряємо умову (4.17). Якщо вона не виконується, переходимо на наступний крок, інакше – на крок 5.

Крок 4. Вилучаємо з множини W ті елементи, які відповідають індексам обмежень, для яких не виконується умова (4.17).

Крок 5. Присвоюємо лічильнику w значення кількості елементів W , тобто $w := w + |W|$.

Крок 6. Формуємо нову систему S_τ , що не містить неактивних обмежень, по елементах множини W .

Виконаємо оцінку роботи алгоритму А3, використовуючи наведений алгоритм та його блок-схему (див. додаток А, рис.А.2). Як і в попередніх алгоритмах, особливу увагу звертаємо на операції множення, ділення, які виконуються повільніше, ніж порівняння, додавання та віднімання.

Алгоритм А3 реалізується на ітераціях $\tau > 2$ розв'язування задачі (4.10)-(4.15) для відкидання обмежень, приєднаних на попередніх ітераціях, при умові, що вони не є активними в точці z^τ і значення цільової функції у достатній мірі покращилось з моменту приєднання цих обмежень. Не обмежуючи загальності міркувань, оцінимо кількість операцій алгоритму А3 на третій ітерації ($\tau = 3$). На першій ітерації система S_1 містить $p + 1 + 2k$ лінійних обмежень. Будемо вважати, що знайдений розв'язок z^1 не задовольняє $2(k - 1)$ обмежень, які приєднуються до системи S_1 за алгоритмом А2. На третій ітерації знову в найгіршому випадку буде приєднано $2(k - 1)$ обмежень.

Отже, коли реалізується алгоритм А3 після знаходження z^3 система містить $p + 1 + 2C_k^1 + 4(k - 1) = p - 1 + 6k$ лінійних обмежень. Серед них перевірка умови (4.16) здійснюється лише для нерівностей, приєднаних на другій ітерації, тобто для $2(k - 1)$ обмежень.

Для обчислення значень $f_{2,j}(z^\tau)$ необхідно виконати: для нерівностей спілки $i = 2 - 4$ операції додавання, для $i = 3 - 6$, для $i = 4 - 8$ і т.д. Утворена послідовність є арифметичною прогресією, сума $(k - 1)$ -го членів якої дорівнює $\frac{4 + 2k}{2}(k - 1) = (k + 2)(k - 1) : k^2$. Аналогічно для кількості операцій множення запишемо послідовність 6, 8, 10, ..., $2(k + 1)$. Всього для $(k - 1)$ спілки їх буде

виконано $(k+4)(k-1): k^2$. Для перевірки умови $f_{2,j}(z^\tau) \leq -\varepsilon$ виконується $2(k-1)$ операцій віднімань і стільки ж порівнянь. Будемо вважати, що умова (4.16) виконується для усіх $2(k-1)$ нерівностей, що перевіряються.

Для перевірки умови (4.17) виконуються наступні операції:

1. Обчислення $F(z^\tau)$, що потребує $(m+1)$ операцій множення та m додавань, здійснюється один раз для всіх $2(k-1)$ перевірок. Таку ж кількість операцій потребує й обчислення значення $F(z^{i-1})$.

2. Для розрахунку $f_{2,j}(z^{2-1})$ потрібно виконати, як описано вище, $(k+4)(k-1): k^2$ операцій множення та $(k+2)(k-1): k^2$ додавання.

Крім того, для порівняння розрахованих величин за умовою $F(z^\tau) \leq F(z^{i-1}) + f_{i,j}(z^{i-1})$ необхідно $2(k-1)$ операцій додавання і стільки ж віднімань та порівнянь.

Таким, чином усього на третій ітерації алгоритм А3 виконує: $(2k^2 + 6k + 2m - 6)$ операцій множення, $(2k^2 + 4k + 2m - 6)$ операцій додавання.

Операцій віднімання та порівняння по $(4k - 4)$.

Можлива ситуація, коли всі $2(k-1)$ нерівності будуть відкинуті за алгоритмом А3. Це більшою мірою залежить від вибору ε . Як показує досвід розв'язування задач [2], ε необхідно вибирати з діапазону $[10^{-2}; 10^{-3}]$. Збільшення величини ε призводить до великого накопичення обмежень на кожному кроці алгоритму, зменшення – до збереження лише активних обмежень. Враховуючи попередні зауваження, будемо вважати, що за алгоритмом А3 на кожній ітерації, починаючи з третьої, відкидаються $2(k-1)$ нерівності, приєднані на попередній ітерації, тобто при кожному зверненні до алгоритму А3 буде виконуватися однакова кількість операцій. Кількість таких

звернень буде на 2 менше, ніж звернень до алгоритму A1, тобто не більше ніж $(2m - 2)$.

Алгоритм A4 реалізує лексикографічну модифікацію двоїстого симплекс-методу. По загальному алгоритму A розв'язування задачі (4.8)-(4.13) алгоритм A4 використовується, як допоміжний, в алгоритмах A5 та A6.

Лексикографічна модифікація двоїстого симплекс-методу реалізована для пошуку лексикографічно максимальної точки опуклого многогранника. При цьому для збереження таблиці T_k l -нормальною при виборі змінної x_l , що вводиться у базис, необхідне виконання умови

$$\frac{R_l}{|x_{kl}|} = \text{lex min} \left\{ \frac{R_j}{|x_{kj}|} \mid j \in N_k, x_{kj} < 0 \right\}, \text{ де } N_k - \text{множина небазисних змінних.}$$

Для знаходження лексикографічно мінімальної точки многогранника недостатньо змінити коефіцієнти цільової функції на протилежні, як у звичайному двоїстому симплекс-методі. Побудова початкового l -псевдоплану забезпечує виконання умови l -антинормальної таблиці T_k .

Означення 4.1. T_k -таблицю назовемо l -антинормальною, якщо перша відмінна від нуля компонента довільного вектора R_j , де $j \in N_k$, є від'ємною.

Для збереження таблиці l -антинормальною при переході від одного l -псевдоплану до іншого, вибираємо змінну x_l , що вводиться в базис, за

$$\text{правилом: } \frac{R_l}{|x_{kl}|} = \text{lex max} \left\{ \frac{R_j}{|x_{kj}|} \mid j \in N_k, x_{kj} < 0 \right\}.$$

На вході в алгоритм задача (4.4), (4.5), (4.6), (4.23), (4.24) або (4.4), (4.5), (4.6), (4.26), (4.27) (див. далі), на виході – розв'язок у вигляді точки $x' = (x_1, \dots, x_k) \in R^k$ та лічильник кількості ітерацій k алгоритму A4. На вході в алгоритм $k = 0$.

Крок 1. Заповнюємо таблицю T_k та перевіряємо, чи є вона допустимою. Якщо так, застосовуємо лексикографічну модифікацію прямого симплекс

методу. Інакше – працює лексикографічно двоїстий симплекс метод, починаючи з кроку 2.

Крок 2. Перевіряємо, чи є таблиця T_k l -нормальною (l -антинормальною для пошуку лексикографічно мінімальної точки). Якщо так, переходимо на крок 5. В іншому разі будуємо l -псевдоплан. Для цього до останнього рядка таблиці T_k додаємо нерівність $\sum_{j \in N_k} x_j \leq M$, або $x_{n+1} = M + \sum_{j \in N_k} (-x_j)$, де M – досить велике додатне число та визначаємо змінну x_k , що виводиться з базису, з умови $k = n+1$, та x_l , що вводиться в базис, за правилом $R_l = \text{lex min} \{R_j \mid j \in N_k\}$ ($R_l = \text{lex max} \{R_j \mid j \in N_k\}$ для пошуку лексикографічно мінімальної точки).

Крок 3. Розраховуємо елементи нової таблиці T_k відносно розв'язувального елемента x_{kl} .

Крок 4. Перевіряємо, чи є таблиця T_k допустимою. Якщо так, задача розв'язана й $x' = (x_1, \dots, x_k)$ – її розв'язок, інакше переходимо на крок 5.

Крок 5. Переходимо до нового l -псевдоплану: змінну x_k для виведення з базису вибираємо з умови $k = \min \{i \mid i = 1, 2, \dots, n; x_{i0} < 0\}$; x_l вводимо в базис за

$$\text{правилом } \frac{R_l}{|x_{kl}|} = \text{lex min} \left\{ \frac{R_j}{|x_{kj}|} \mid j \in N_k, x_{kj} < 0 \right\} \quad \left(\frac{R_l}{|x_{kl}|} = \text{lex max} \left\{ \frac{R_j}{|x_{kj}|} \mid j \in N_k, x_{kj} < 0 \right\} \right)$$

для знаходження лексикографічно мінімальної точки).

Крок 6. Перевіряємо, чи існують $x_{kj} < 0$. Якщо ні, виходимо з алгоритму А4, задача розв'язку не має. Інакше $k := k + 1$ та переходимо на крок 3.

Враховуючи, що вихідна задача, наприклад вигляду (4.4), (4.5), (4.6), (4.23), (4.24), містить систему нерівностей (4.5), таблиця T_k на кроці 1 алгоритму не є допустимою, а отже, лексикографічна модифікація прямого симплекс методу не реалізується для розглядуваної задачі.

Виконаємо оцінку роботи алгоритму А4, використовуючи наведений алгоритм та його блок-схему (див. додаток А, рис.А.3). Оцінку операцій

зробимо в розрахунку на одну ітерацію алгоритму, тобто перехід від одного псевдоплану до іншого.

Алгоритм А4 реалізується з підключенням алгоритму А2, що впливає на вимірність розв'язуваної задачі та кількість ітерацій.

Нехай вихідна задача, наприклад, вигляду (4.4), (4.5), (4.6), (4.23), (4.24), містить $p + \rho + 2 \sum_{i=1}^k C_k^i$ лінійних обмежень. Згідно з алгоритмом А2 кількість обмежень зменшуємо до $p + \rho + 2k$, де $\rho \in J_k$. Тоді всього змінних у задачі буде $p + \rho + 2k + m$, серед яких m – небазисні та $(\rho - 1)$ штучна. Для заповнення таблиці T_k необхідно записати цільову функцію та всі базисні змінні через небазисні $(-x_j)$, де $j \in N_k$. Кількість операцій для запису цільової функції залежить від кількості штучних змінних, оскільки кожна $(\rho - 1)$ штучна змінна, що входить в цільову функцію з коефіцієнтом $(-M)$, повинна бути виражена через небазисні змінні. Для цього необхідно виконати $(\rho - 1)$ додавання або віднімання та $2(\rho - 1)$ операцій множення. Інші елементи таблиці отримуємо зчитуванням коефіцієнтів при змінних вихідної задачі та переприсвоюванням їх значень. Для перевірки таблиці T_k на допустимість виконується $(p + \rho + 2k)$ операцій порівняння x_{i0} з нулем, оскільки $x_{i0} = 0$ для небазисних змінних. Перевірка, чи є таблиця T_k l -нормальною (l -антинормальною), потребує не більше $m(p + \rho + 2k + 1 + m)$ операцій порівняння. Припустимо, що таблиця T_k не є l -нормальною (l -антинормальною), тоді побудова l -псевдоплану призводить до розширення таблиці T_k за рахунок додаткового обмеження. Вибір змінної x_k , що виводиться з базису, визначається однозначно і потребує однієї операції, а для введення змінної x_l в базис потрібно не більше $m(p + \rho + 2k + 1 + m)$ операцій порівняння.

Заповнення елементів нової таблиці здійснюється з використанням операцій віднімання, множення та ділення, причому для всіх елементів, крім

останнього рядка, який викреслюється з таблиці. Розрахунок елементів прямого стовпця l потребує $(p + r + 2k + m + 1)$ операцій ділення. Для інших m стовпців по $(p + r + 2k + m + 1)$ рядках виконується по $m(p + r + 2k + m + 1)$ операцій множення, ділення та віднімання.

Для перевірки таблиці T_k на допустимість виконується не більше, ніж $(p + r + 2k + m)$ операцій порівняння x_{i0} з нулем, причому одночасно фіксується найменший індекс k змінної, що виводиться з базису. Для визначення змінної x_l , що вводиться в базис, необхідно виконати не більше, ніж m^2 операцій ділення. Для перевірки умови наявності від'ємних компонент x_{kj} потрібно m операцій порівняння. Збільшуємо номер ітерації алгоритму A4 та переходимо до нового l -псевдоплану.

Таким чином, до ітераційного процесу в алгоритмі A4 виконується: $(r - 1)$ додавань або віднімань, $2(r - 1)$ операцій множення та не більше $m(m + 1)$ операцій порівняння. Кількість операцій для однієї ітерації алгоритму A4 наведена в таблиці 4.2.

Таблиця 4.2

Кількість операцій однієї ітерації алгоритму A4

Операції	Кількість операцій
множення	$m(p + r + 2k + m + 1)$
ділення	$p + r + 2k + 1 + mr + mr + 2mk + 2m^2 + 2m$
віднімання	$m(p + r + 2k + m + 1)$
порівняння	$p + r + 2k + 2m$

Кожний перехід від одного псевдоплану до наступного складає ітерацію. При застосуванні описаного алгоритму ми не можемо повернутися до вже пройденого раз базису за рахунок монотонного спадання цільової функції в

задачі на лексикографічний максимум (зростання цільової функції в задачі на лексикографічний мінімум). Таким чином, кількість ітерацій необхідних для розв'язування невивірженої задачі наперед обмежена загальною кількістю базисів, що не перевищує $C_{p+\rho+2k+m}^{p+\rho+2k}$, де $p+\rho+2k+m$ – загальна кількість змінних розв'язуваної задачі в канонічному вигляді, серед яких $(\rho-1)$ штучна, $p+\rho+2k$ – кількість обмежень.

Зауваження. При виведенні штучної змінної з базису, вона не може бути знову включена в базис, тому кількість можливих базисів значно менше величини $C_{p+\rho+2k+m}^{p+\rho+2k}$.

Отже, верхня оцінка кількості операцій алгоритму A4 дорівнює добутку кількості ітерацій на значення кількості операцій однієї ітерації, оскільки вимірність задачі до отримання оптимального розв'язку, що задовольняє задану систему обмежень, не змінюється. Однак згідно з алгоритмом A2, шляхом додавання до системи обмежень нерівностей, що не виконуються в знайденій точці, вимірність задачі зростає, що впливає суттєво на кількість ітерацій алгоритму A4 та кількість операцій в одній ітерації.

Алгоритм A5 реалізує задачу пошуку представника комбінаторного класу найближчого зліва до $\bar{V}$, представником якого є $\bar{x}$, або просто $\lfloor V_1 \rfloor$ -задачу.

На вході в алгоритм A5 точка $\bar{x}$, що є представником класу $\bar{V}$, на виході – найближчий зліва до $\bar{V}$ комбінаторний клас $\lfloor V_1 \rfloor$ з представником $\lfloor x_1 \rfloor$. На вході в алгоритм лічильник кількості ітерацій $h_l = 0$.

Крок 1. Перевіряємо, чи утворює $\bar{V}$ некомбінаторний клас. Якщо так, $\rho := \rho(\bar{V})$, інакше $\rho := k$.

Крок 2. Знаходимо $e_i = \max \{e_i \mid e_i \in S(G), e_i < \bar{x}_\rho, (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\rho-1}, e_i) \in E^\rho\}$. Якщо такого $e_i \in S(G)$ не існує, переходимо на крок 3, інакше на крок 4.

Крок 3. Якщо $\rho > 1$, зменшуємо ρ на одиницю і переходимо на крок 2, інакше задача розв'язку не має.

Крок 4. Знаходимо лексикографічно максимальну точку многогранника, визначеного умовами (4.4)-(4.6) та

$$x_j = \bar{x}_j \quad \forall j \in J_{\rho-1}, \quad (4.23)$$

$$x_\rho \leq e_i. \quad (4.24)$$

Для цього розв'яжемо

$$F(x') = \underset{x \in R^k}{\text{lex max}} x_\rho, \quad x' = \underset{x \in R^k}{\text{arg lex max}} x_\rho, \quad (4.25)$$

при умовах (4.4)-(4.6), (4.23), (4.24) алгоритмом А4.

Крок 5. Приймаємо $h_l := h_l + 1$. Якщо задача (4.4)-(4.6), (4.23)-(4.25) не має розв'язку, то переходимо на крок 3, інакше нехай ρ – ранг класу, що визначається розв'язком (4.4)-(4.6), (4.23)-(4.25).

Крок 6. Якщо $\rho = k + 1$, то задача розв'язана і $\lfloor x_1 \rfloor$ – шукана точка (визначає клас $\lfloor V_1 \rfloor$), інакше повертаємося на крок 2.

Оцінимо роботу алгоритму А5. Оцінку операцій зробимо в розрахунку на одну ітерацію алгоритму.

Визначення ρ – рангу класу $\bar{V}$, або індексу компоненти точки $\bar{x} = x^\tau = (x_1^\tau, x_2^\tau, \dots, x_k^\tau)$, що не належить множині $G = \{g_1, g_2, g_3, \dots, g_\eta\}$, виконується по алгоритму, програмна реалізація якого представлена в додатку Б.1. Для реалізації цього алгоритму виконуються лише операції порівняння. Сортування компонент точки $\bar{x}$ за значенням у порядку зростання потребує у найгіршому випадку k^2 операцій. Цикл *while do* по компонентам точки $\bar{x}$ виконується не більше ніж k раз, що містить $2k$ операцій порівняння. Цикл *while do* по елементам мультимножини G виконується не більше ніж η раз, що

містить 3η операцій порівняння. Таким чином, процедура визначення ρ потребує всього $k^2 + 2k + 3\eta$ операцій порівняння.

Із кроку 2 алгоритму A5 починається основний цикл *repeat* алгоритму. Програмна реалізація кроку 2 алгоритму A5 представлена в додатку Б.2. Сортювання k перших компонент точки $\bar{x}$ за індексом у порядку зростання потребує у найгіршому випадку k^2 операцій. Знаходження максимального елемента e_i , що задовольняє умову $e_i < \bar{x}_\rho$ потребує не більше ніж $(\eta - 1)$ операцій порівняння. Цикл, що забезпечує виконання умови $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\rho-1}, e_i) \in E^\rho$, може виконуватися не більше ніж $(\rho - 1)$ разів, в кожному з яких в середньому виконується 4 операції порівняння. Отже, всього процедура знаходження $e_{\max}$ потребує $k^2 + (\eta - 1) + 4(\rho - 1)$ операцій порівняння. Далі в основному циклі йде на одну ітерацію 3 перевірки умови та звернення до алгоритму A4 і кроку 2 алгоритму A5, описаних та проаналізованих вище. Таким чином, одна ітерація алгоритму A5 містить $[3k^2 + 4k + 7\eta + 4\rho - 2 + R]$, де R – кількість операцій однієї ітерації алгоритму A4, представлена в таблиці 4.2, помножена на кількість можливих ітерацій $C_{p+\rho+2k+m}^{p+\rho+2k}$.

Кількість ітерацій алгоритму A5 визначається послідовністю знайдених розв'язків задач вигляду (4.4)-(4.6), (4.23)-(4.25) аж до отримання точки, що є представником комбінаторного класу. Кількість звернень до алгоритму A4, як показують числові експерименти, залежить від багатьох факторів, а саме, від кількості елементів мультимножини G та її основи, від характеру розташування додаткових лінійних обмежень відносно опуклої оболонки множини розміщень. За результатами тестування кількість ітерацій алгоритму A5 для задач вимірностей від 15 до 33 змінюється від 1 до 10000. Найбільша кількість ітерацій зафіксована для випадку, коли допустима область вже не містить комбінаторних класів, і алгоритм намагаючись знайти його, перебирає усі можливі варіанти, пробігаючи тисячі раз по циклу.

Алгоритм А6 реалізує задачу пошуку представника комбінаторного класу найближчого справа до $\bar{V}$, представником якого є $\bar{x}$, або просто $\lceil V_1 \rceil$ -задачу.

На вході в алгоритм А6 точка $\bar{x}$, що є представником класу $\bar{V}$, на виході – найближчий справа до $\bar{V}$ комбінаторний клас $\lceil V_1 \rceil$ з представником $\lceil x_1 \rceil$. На вході в алгоритм лічильник кількості ітерацій $h_r = 0$.

Крок 1. Перевіряємо, чи утворює $\bar{V}$ некомбінаторний клас. Якщо так, $\rho := \rho(\bar{V})$, інакше $\rho := k$.

Крок 2. Знаходимо $e_i = \min \{e_i \mid e_i \in S(G), e_i > \bar{x}_\rho, (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{\rho-1}, e_i) \in E^\rho\}$.

Якщо такого $e_i \in S(G)$ не існує, переходимо на крок 3, інакше на крок 4.

Крок 3. Якщо $\rho > 1$, зменшуємо ρ на одиницю і переходимо на крок 2, інакше задача розв'язку не має.

Крок 4. Знаходимо лексикографічно максимальну точку многогранника, визначеного умовами (4.4)-(4.6) та

$$x_j = \bar{x}_j \quad \forall j \in J_{\rho-1}, \quad (4.26)$$

$$x_\rho \geq e_i. \quad (4.27)$$

Для цього розв'яжемо

$$F(x') = \underset{x \in R^k}{\text{lex min}} x_\rho, \quad x' = \arg \underset{x \in R^k}{\text{lex min}} x_\rho, \quad (4.28)$$

при умовах (4.4), (4.5), (4.6), (4.26), (4.27) алгоритмом А4.

Крок 5. Приймаємо $h_r := h_r + 1$. Якщо задача (4.4)-(4.6), (4.26)-(4.28) не має розв'язку, то переходимо на крок 3, інакше нехай ρ – ранг класу, що визначається розв'язком (4.4), (4.5), (4.6), (4.26)-(4.28).

Крок 6. Якщо $\rho = k + 1$, то задача розв'язана і $\lceil x_1 \rceil$ – шукана точка (визначає клас $\lceil V_1 \rceil$), інакше повертаємося на крок 2.

Аналізуючи алгоритм А6 та фрагмент його програмної реалізації (див. додаток Б.3), бачимо, що ефективність роботи алгоритму А6 така ж, як і алгоритму А5.

Перейдемо до розгляду алгоритмів перегляду комбінаторних класів. При цьому, як уже зазначалося, з розгляду виключаються точки, які надають цільовій функції значення менше F^* , де F^* було одержано на одній із попередніх ітерацій. Це означає, що відкидаються точки, які не задовольняють нерівність

$$\frac{\sum_{j=1}^m c_j t_j + c_0}{\sum_{j=1}^m d_j t_j + d_0} \geq F^*. \quad (4.29)$$

Дробово-лінійну нерівність (4.29) можна записати у вигляді лінійної нерівності. Виконаємо такі перетворення:

$$\frac{\sum_{j=1}^m c_j t_j + c_0}{\sum_{j=1}^m d_j t_j + d_0} - F^* \geq 0,$$

$$\frac{\sum_{j=1}^m c_j t_j + c_0 - F^* \left(\sum_{j=1}^m d_j t_j + d_0 \right)}{\sum_{j=1}^m d_j t_j + d_0} \geq 0. \text{ Оскільки, за припущенням, } \sum_{j=1}^m d_j t_j + d_0 > 0, \text{ то}$$

перетворена нерівність запишеться так

$$\sum_{j=1}^m c_j t_j + c_0 - F^* \left(\sum_{j=1}^m d_j t_j + d_0 \right) \geq 0. \quad (4.30)$$

Остання нерівність є лінійною.

Алгоритм А7 перегляду комбінаторних класів у порядку спадання, починаючи з точки $\lfloor x_i \rfloor$.

На вході в алгоритм А7 точка $\lfloor x_1 \rfloor$, що є представником класу $\lfloor V_1 \rfloor$, на виході – найменший в лексикографічному порядку комбінаторний клас $\lfloor V_i \rfloor$ з представником $\lfloor x_i \rfloor$, компоненти якого задовольняють нерівність (4.30).

Крок 1. Нехай $i=1$. Додаємо до системи обмежень (4.4)-(4.6) нерівність (4.30), де $F^* = F(\lfloor x_1 \rfloor)$.

Крок 2. Розв'язуємо задачу (4.4)-(4.6), (4.23)-(4.25) (4.30) пошуку комбінаторного класу, найближчого зліва до λ -класу з представником $\lfloor x_i \rfloor$.

Крок 3. Якщо задача не має розв'язку, то комбінаторного класу, розташованого лівіше $\lfloor V_i \rfloor$, не існує, приймаємо $x^* = \lfloor x_i \rfloor$, отримаємо $\langle F(x^*), x^* \rangle$ – результат перебору, інакше нехай $\lfloor x_{i+1} \rfloor$ (визначає клас $\lfloor V_{i+1} \rfloor$) – розв'язок задачі.

Крок 4. Покладаємо $F^* = F(\lfloor x_{i+1} \rfloor)$, $x^* = \lfloor x_{i+1} \rfloor$ і змінюємо нерівність (4.30).

Крок 5. Збільшуємо i на одиницю і повертаємося на крок 2.

Оцінимо роботу алгоритму А7. Оцінку операцій зробимо в розрахунку на одну ітерацію алгоритму.

При формуванні нерівності (4.30) необхідно розрахувати значення цільової функції в точці, що є представником комбінаторного класу, починаючи з якого ведемо перегляд. Для цього, враховуючи вимірність простору R^m , необхідно виконати $2m$ раз операцію множення, $2(m-1)$ раз операцію додавання, 1 операцію порівняння знаменника з нулем і, якщо умова не виконується, 1 ділення чисельника на знаменник.

На наступному кроці розв'язування задачі на множині (4.4)-(4.6), (4.30) реалізується з підключенням алгоритму А5, який в свою чергу звертається до

алгоритму А4. Алгоритм А5 проаналізований вище для даної задачі та оцінена кількість операцій, алгоритм А4 застосовується до задачі вигляду (4.4)-(4.6), (4.23)-(4.25), (4.30), яка містить $p + \rho + 2 \sum_{i=1}^k C_k^i + 1$ лінійних обмежень. Згідно з алгоритмом А2 кількість обмежень зменшуємо до $p + \rho + 2k + 1$, де $\rho \in J_k$. Тоді всього змінних у задачі буде $p + \rho + 2k + m + 1$, серед яких m – небазисні та $(\rho - 1)$ штучна. Аналогічно для задачі (4.4)-(4.6), (4.23)-(4.25), (4.30) отримуємо кількість операцій для однієї ітерації алгоритму А4, які наведені в таблиці 4.3.

Кількість ітерацій алгоритму А4 наперед обмежена загальною кількістю базисів, що не перевищує $C_{p+\rho+2k+m+1}^{p+\rho+2k+1}$, де $p + \rho + 2k + m + 1$ – загальна кількість змінних розв’язуваної задачі в канонічному вигляді, серед яких $(\rho - 1)$ штучна, $p + \rho + 2k + 1$ – кількість обмежень.

Таблиця 4.3

**Кількість операцій однієї ітерації для розв’язування задачі (4.4)-(4.6),
(4.23)-(4.25), (4.30) алгоритмом А4**

Операції	Кількість операцій
множення	$m(p + \rho + 2k + m + 2)$
ділення	$(p + \rho + 2k + m + 2) + m(p + \rho + 2k + 2) + 2m^2$
віднімання	$m(p + \rho + 2k + m + 2)$
порівняння	$p + \rho + 2k + 2m + 1$

Далі по алгоритму А7 йде перевірка умови, операції присвоєння й знову звернення до алгоритму А5 у циклі. Враховуючи, що алгоритм А7 реалізує перегляд комбінаторних класів у порядку спадання, кількість ітерацій цього алгоритму наперед обмежена зверху кількістю комбінаторних класів. Оскільки мультимножина G , що визначає загальну множину розміщень $E_{\eta\eta}^k(G)$, містить елементи, що повторюються, тому кількість комбінаторних класів визначимо з

умови n^k – кількість розміщень з n різних елементів з необмеженими k повтореннями.

Алгоритм А8 перегляду комбінаторних класів у порядку зростання, починаючи з точки $\lceil x_i \rceil$ [107].

Крок 1. Нехай $i=1$. Додаємо до системи обмежень (4.4), (4.5), (4.6) нерівність (4.30), де $F^* = F(\lceil x_1 \rceil)$.

Крок 2. Розв'язуємо (4.4)-(4.6), (4.26)-(4.28), (4.30) – задачу пошуку комбінаторного класу, найближчого справа до λ -класу з представником $\lceil x_i \rceil$.

Крок 3. Якщо задача не має розв'язку, то комбінаторного класу, розташованого правіше $\lceil V_i \rceil$, не існує і, покладаючи $x^* = \lceil x_i \rceil$, маємо $\langle F(x^*), x^* \rangle$ – результат перебору. Інакше нехай $\lceil x_{i+1} \rceil$ (визначає клас $\lceil V_{i+1} \rceil$) – розв'язок задачі.

Крок 4. Покладаємо $F^* = F(\lceil x_{i+1} \rceil)$, $x^* = \lceil x_{i+1} \rceil$ і змінюємо нерівність (4.30).

Крок 5. Збільшуємо i на одиницю і повертаємося на крок 2.

Проаналізувавши алгоритм А8, легко бачити, що оцінка роботи його така ж, як і алгоритму А7.

Таким чином, можна запропонувати такий **алгоритм А** розв'язування задачі (4.1)-(4.4), блок-схема якого представлена в додатку А, рис. А.4.

Крок 1. Здійснюємо релаксацію задачі (4.1)-(4.4) та застосовуємо до останньої перетворення (4.7), отримаємо задачу (4.8)-(4.13), яку розв'язуємо модифікованим симплекс методом (алгоритм А1 з підключенням алгоритмів А2 та А3). Нехай $\bar{y}$ – розв'язок задачі (4.8)-(4.13), а $\bar{x}$ – розв'язок задачі (4.1), (4.2), (4.4)-(4.6).

Крок 2. Якщо $\bar{x} \in E_{\eta n}^k(G)$, то покладаємо $x^* = \bar{x}$, (4.1)-(4.4) розв'язана і $\langle F(x^*), x^* \rangle$ – її розв'язок, інакше переходимо на крок 3.

Крок 3. Звернення до алгоритмів A5 та A6, що реалізують задачі пошуку представників комбінаторних класів найближчого відповідно зліва та справа до $\bar{V}$, представником якого є $\bar{x}$. На вході в алгоритми A5 та A6 точка $\bar{x}$, що є представником класу $\bar{V}$, на виході – найближчий відповідно зліва та справа до $\bar{V}$ комбінаторний клас.

Крок 4. Якщо жодна із задач пошуку представників комбінаторних класів найближчого відповідно зліва та справа до $\bar{V}$ не має розв'язку, то не має розв'язку і (4.1)-(4.4), інакше переходимо на крок 5.

Крок 5. Якщо задача, що реалізується алгоритмом A5, не має розв'язку, то звертаємося до алгоритму A8, що реалізує перегляд комбінаторних класів у порядку лексикографічного зростання, починаючи з $\lceil x_1 \rceil$, та переходимо на крок 14.

Крок 6. Якщо задача, що реалізується алгоритмом A6, не має розв'язку, то звертаємося до алгоритму A7, що реалізує перегляд комбінаторних класів, починаючи з $\lfloor x_1 \rfloor$, у порядку лексикографічного спадання та переходимо на крок 14.

Крок 7. Якщо $\lfloor x_1 \rfloor$, $\lceil x_1 \rceil$ – розв'язки задач за алгоритмами A5 і A6 відповідно і $\lfloor x_1 \rfloor$ кращий допустимий розв'язок, ніж $\lceil x_1 \rceil$ (для задач максимізації $F(\lfloor x_1 \rfloor) > F(\lceil x_1 \rceil)$ або $F(\lfloor x_1 \rfloor) = F(\lceil x_1 \rceil)$ і $\lfloor x_1 \rfloor \neq \lceil x_1 \rceil$), то переходимо на крок 8, інакше на крок 11.

Крок 8. Звернення до алгоритму A7, що реалізує перегляд класів у порядку лексикографічного спадання, починаючи з $\lfloor x_1 \rfloor$. Нехай $\langle F(x^*), x^* \rangle$ – пара, отримана у результаті перегляду.

Крок 9. Звернення до алгоритму A6, що розв'язує задачу вигляду (4.28) на множині, визначеній умовами (4.4)-(4.6), (4.26)-(4.27). Якщо вона не має розв'язку, переходимо на крок 14, інакше $\lceil x_1 \rceil$ – розв'язок задачі (4.4)-(4.6), (4.26)-(4.28).

Крок 10. Звернення до алгоритму A8, що реалізує перегляд комбінаторних класів у порядку лексикографічного зростання, починаючи з $\lceil x_1 \rceil$, де $F^* = F(\lceil x_1 \rceil)$, і перехід на крок 14.

Крок 11. Звернення до алгоритму A8, що реалізує перегляд комбінаторних класів у порядку лексикографічного зростання, починаючи з $\lceil x_1 \rceil$. Нехай $\langle F(x^*), x^* \rangle$ – пара, отримана у результаті перегляду.

Крок 12. Звернення до алгоритму A5, що розв'язує задачу вигляду (4.25) на множині, визначеній умовами (4.4)-(4.6), (4.23)-(4.24). Якщо вона не має розв'язку, переходимо на пункт 14, інакше $\lfloor x_1 \rfloor$ – розв'язок на множині (4.4)-(4.6), (4.23)-(4.24).

Крок 13. Звернення до алгоритму A7, що реалізує перегляд комбінаторних класів у порядку лексикографічного спадання, починаючи з $\lfloor x_1 \rfloor$, де $F^* = F(\lfloor x_1 \rfloor)$, і перехід на крок 14.

Крок 14. Розв'язком задачі (4.1)-(4.4) є пара $\langle F(x^*), x^* \rangle$.

Таким чином, побудовано алгоритм розв'язування умовної задачі з дробово-лінійною функцією на розміщеннях шляхом розвинення алгоритму розв'язування лінійних задач на розміщеннях. Даний алгоритм може бути застосовним як для розв'язування частково комбінаторних задач так і для повністю комбінаторних, як для умовних так і для безумовних задач з дробово-лінійною функцією цілі.

4.4. Числові експерименти та приклади за методом побудови лексикографічної еквівалентності

Розроблений метод реалізовано в середовищі Delphi з використанням мови Object Pascal. Програмний комплекс включає 8 модулів: uDSxM, uGlobal,

uMSxM, uSxBooleanUtils, uWxEquations, uWxInfo, uWxResult, uWxTask.
Основні модулі такі:

- uWxTask – головний модуль, що відповідає за розрахунки задачі, містить основні функції та процедури для обчислень.
- uDSxM – модуль реалізації лексикографічної модифікації двоїстого симплекс-методу.
- uMSxM – модуль модифікованого симплекс-методу.
- uWxResult – допоміжний модуль для обробки результатів розрахунків.
- uWxInfo – модуль для відображення результатів роботи програми.

Практична ефективність алгоритму оцінювалася шляхом числових експериментів. Програма отримувала вхідні дані у вигляді текстових файлів, де задавалися мультимножина, коефіцієнти цільової функції та додаткових обмежень. Для генерації цих даних використовувалася стандартна процедура мови Pascal для створення псевдовипадкових чисел із рівномірним розподілом.

Задачі генерувалися випадковим чином: елементи, що утворюють загальну множину розміщень, визначалися як рівномірно розподілені числа на інтервалах $[1;10]$, $[1;20]$, $[1;30]$, $[1;40]$, $[1;50]$, $[1;100]$. Аналогічним чином генерувалися коефіцієнти цільової функції та додаткових лінійних обмежень.

Основними характеристиками алгоритму, які аналізувалися в експериментах, були:

- Час роботи програми, що обчислювався як різниця між системним часом на початку й наприкінці виконання програми.
- Кількість ітерацій, необхідних для досягнення розв'язку.

У деяких випадках задачі, згенеровані випадковим чином, вирішувалися вже на першому кроці, коли алгоритм завершував роботу після перевірки належності розв'язку множині розміщень. Дані про мінімальний час і кількість ітерацій у таких випадках не враховувалися.

Для аналізу результатів застосовувалася програма Curve Expert 1, яка дозволяє будувати залежності між параметрами за допомогою різних моделей регресії.

Експерименти проводилися на задачах евклідової комбінаторної оптимізації з дробово-лінійною функцією цілі, включаючи:

- повністю комбінаторні задачі з додатковими лінійними обмеженнями;
- частково комбінаторні задачі з додатковими лінійними обмеженнями.

При проведенні експериментів фіксувалися наступні величини:

- η – кількість елементів мультимножини G ;
- n – кількість різних елементів у G ;
- l – інтервал, з якого генеруються елементи мультимножини G ;
- τ – кількість викликів алгоритму A1;
- q – кількість приєднаних обмежень згідно МППО (алгоритму A2);
- w – кількість відкинутих обмежень за алгоритмом A3;
- β – кількість „малих” ітерацій (загальна кількість ітерацій в алгоритмах A5, A6);
- γ – кількість „великих” ітерацій (загальна кількість викликів алгоритмів A5, A6 в алгоритмах A7, A8);
- T – час розв’язування задачі (в форматі хв.: сек.);
- x^* – екстремаль; $F(x^*)$ – значення цільової функції у точці x^* .

Перша серія експериментів складалася з 110 задач при збільшенні вимірності простору m від 15 до 33 і сталій кількості обмежень $p = 2$.

Мінімальний зафіксований час розв’язування задач менше 1 сек., максимальний час розв’язування задач вимірності $m \leq 22$ складає 8:18, для $23 < m \leq 33$ час $T = 48:50$. При перевищенні часу розв’язування задачі більше 1,5 годин здійснювалася примусова зупинка (із 110 задач 7 задач вимірності $23 < m \leq 33$ було зупинено).

У таблиці 4.4. наведені результати в процентному співвідношенні задач, розв'язаних протягом певного проміжку часу (час записано в форматі хв.:сек.).

Таблиця 4.4

Залежність часу розв'язування задачі від вимірності

Вимірність	Кількість задач	Час розв'язування		Відсоток задач, розв'язаних за певний проміжок часу					Відсоток задач, не розв'язаних за 1,5 год.
		середній	максимальний	до 1 сек.	від 1 сек. до 1 хв.	от 1 до 5 хв.	от 5 до 15 хв.	от 15 хв. до 1 год.	
15	10	0:03	0:06	10%	90%	-	-	-	-
16	5	0:09	0:32	20%	80%	-	-	-	-
17	5	0:9	0:28	-	100%	-	-	-	-
18	5	0:13	0:37	20%	80%	-	-	-	-
19	5	1:13	3:56	-	60%	40%	-	-	-
20	5	0:45	1:53	80%	20%	-	-	-	-
21	5	1:56	8:18	-	60%	20%	20%	-	-
22	5	0:14	0:30	-	100%	-	-	-	-
23	5	1:55	5:54	-	40%	20%	20%	-	20%
24	5	4:14	11:44	-	40%	20%	40%	-	-
25	5	6:21	27:26	-	20%	60%		20%	-
26	5	2:47	5:07	-	-	60%	20%	-	20%
27	5	5:58	13:56	-	20%	40%	40%	-	-
28	5	2:43	11:06	-	60%	20%	20%	-	-
29	5	1:36	3:13	-	40%	40%		-	20%
30	10	4:13	22:24	-	40%	40%	10%	10%	-
31	5	26:56	48:50	-	-	-	20%	80%	-
32	10	3:15	9:42	-	40%	20%	10%	-	30%
33	5	2:55	6:18	-	20%	40%	20%	-	20%

Як видно з таблиці 4.4, незалежно від вимірності задачі певна кількість задач розв'язується менше 1 хв. Однак максимальний час розв'язування задач зі збільшенням їх вимірності зростає. Залежність часу розв'язування задач від їх вимірності не представлена, оскільки не було побудовано жодної лінії регресії з коефіцієнтом кореляції більшим 0,5.

За результатами експериментів досліджувалася залежність часу розв'язування задач T від кількості «малих» ітерацій β . Найкраща (по коефіцієнту кореляції $r=0,94$) лінія регресії на рис. 4.1 – це поліном третього степеня $2,17 \cdot 10^{-12}x^3 - 2,94 \cdot 10^{-7}x^2 + 0,02x - 34,07$.

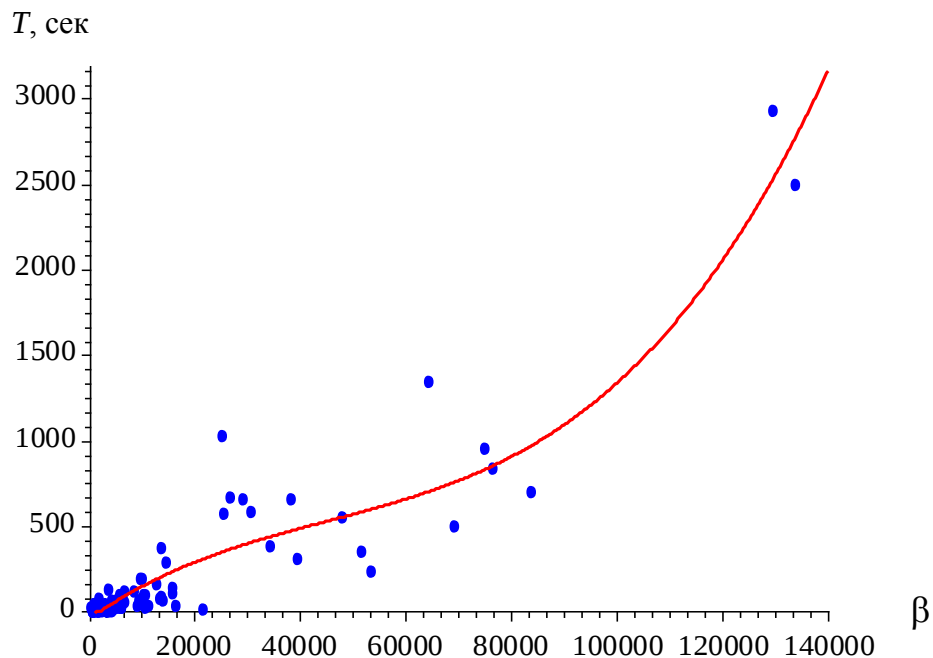


Рис. 4.1 Графік залежності часу розв'язування задачі T від кількості «малих» ітерацій β

Досліджувалася також залежність часу розв'язування задач T від кількості «великих» ітерацій γ . На рис. 4.2 представлено графік лінії регресії $0,12x^{0,28}$ (коефіцієнт кореляції $r=0,82$).

Зауважимо, що, чим більше значення параметра l , тобто, чим більший інтервал, з якого генеруються елементи мультимножини G , тим більша

кількість звернень до алгоритму A1 і кількість „малих” ітерацій в методі побудови лексикографічної еквівалентності.

Приклади розв’язування задач цієї серії представлені в додатку В (приклади В.1-В.7).

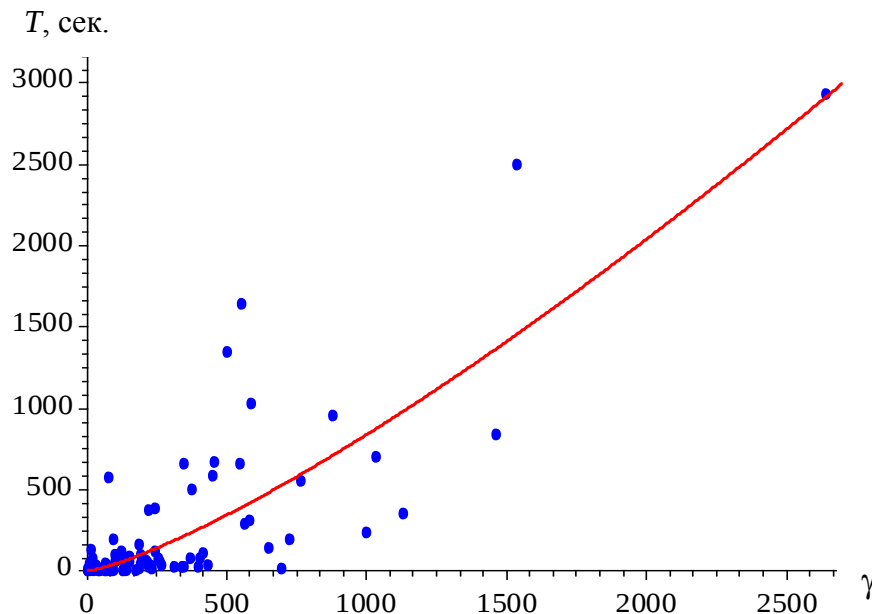


Рис. 4.2 Графік залежності часу розв’язування задачі T від кількості „великих” ітерацій γ

Друга серія розрахунків складалася з 42 тестових задач при послідовному збільшенні кількості додаткових лінійних обмежень від 3 до 13, сталій вимірності простору $m=20$ та кількості елементів мультимножини $\eta=100$ (див., наприклад, додаток В, приклади В.8-В.10).

За результатами експериментів досліджувалася залежність середнього часу розв’язування задач T_c від кількості додаткових лінійних обмежень p . Найкраща (по коефіцієнту кореляції $r=0,95$) лінія регресії $y = \frac{a}{1 + be^{-cx}}$, де $a = -19,773531$, $b = -1,3865768$, $c = 0,023899498$, представлена на рис. 4.3. Кількість додаткових лінійних обмежень та їх розташування є суттєвим фактором, що впливає на час розв’язування задачі. Наявність додаткових обмежень зменшує область

допустимих розв'язків задачі, а тому в деяких задачах час рахунку менший, ніж у задачах такої ж вимірності з меншою кількістю обмежень. Однак в той же час спостерігається тенденція до збільшення часу розв'язування задачі зі збільшенням кількості додаткових обмежень, що пояснюється характером розташування цих обмежень відносно допустимої області, що їх не містить, та збільшенням вимірності задачі, що розв'язується.

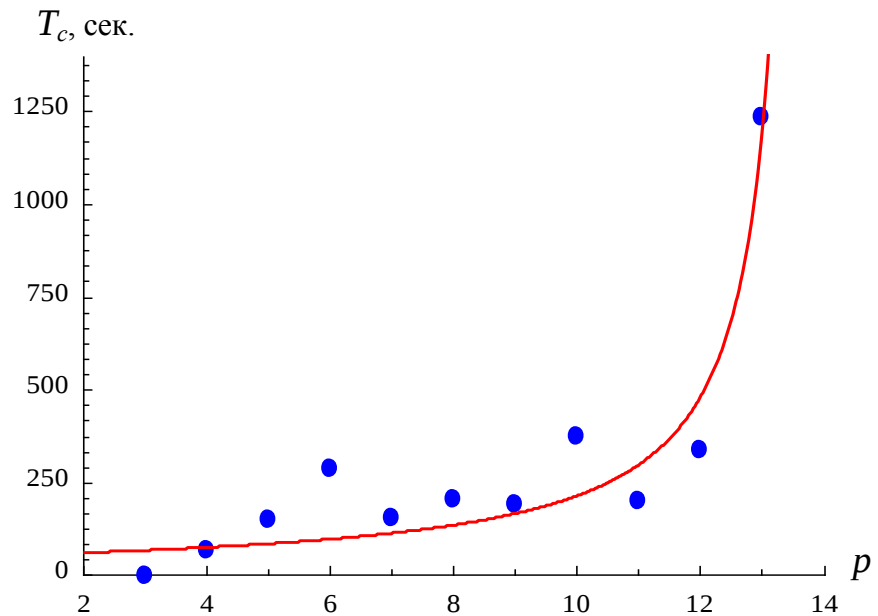


Рис. 4.3 Графік залежності середнього часу T_c розв'язування задач від кількості додаткових обмежень p

Досліджувалася також залежність середньої кількості „великих” ітерацій γ від кількості додаткових обмежень. Аналізуючи рис. 4.4, де представлено графік лінії регресії $3,25x^3 - 92,13x^2 + 802,65x - 1575,3$ (коефіцієнт кореляції $r=0,83$) спостерігається „нерегулярність” поведінки алгоритмів А7, А8, що пояснюється також впливом інших факторів (характером розташування обмежень відносно допустимої області, кількістю елементів мультимножини та її основи та ін.) на середню кількість „великих” ітерацій.

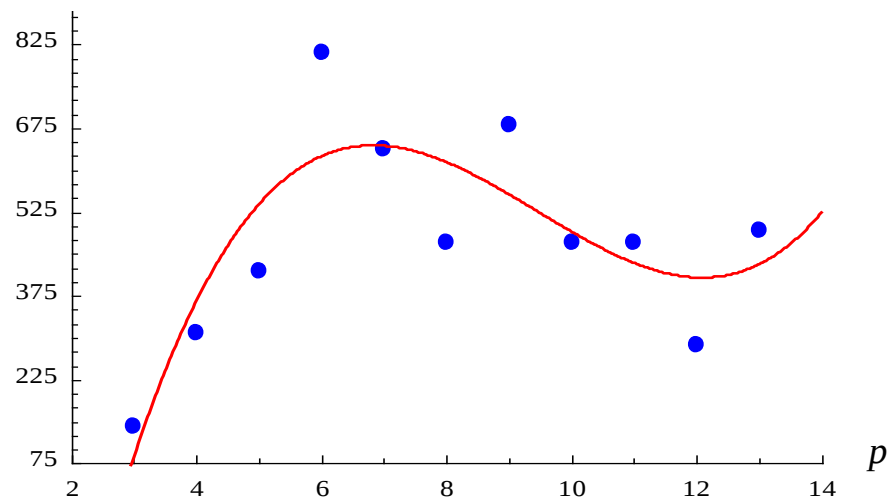


Рис. 4.4 Графік залежності середньої кількості великих ітерацій від кількості додаткових обмежень p

Третя серія розрахунків складалася з 116 задач при значеннях вимірності від 20 до 24 та кількості додаткових лінійних обмежень від 4 до 8. Розрахунки проводилися для елементів мультимножини G , які вибиралися як випадкові рівномірно розподілені числа на інтервалі $l = [1; 50]$. Результати експериментів наведено в таблиці 4.5.

Як видно з таблиці 4.5, кількість додаткових обмежень та їх розташування є суттєвим фактором, що впливає на час рахунку та отримання розв'язку. Наявність додаткових обмежень зменшує область допустимих розв'язків задачі, а отже, зменшується кількість комбінаторних класів, що переглядаються алгоритмами А7 та А8 (задачі вимірностей 20, 21, крім випадку при $p=8$). Однак одночасно росте вимірність задачі, що суттєво впливає на час рахунку алгоритмом А1.

Таблиця 4.5

**Залежність середньої кількості ітерацій та середнього часу
розв'язування задач від вимірності та кількості додаткових обмежень**

Вимірність задачі, m	Кількість задач	Кількість обмежень p	Кількість малих ітерацій, β	Кільк. великих ітерацій, γ	Середній час (хв.: сек.)	Максимальний час (хв.: сек.)
20	2	4	27875	311	2:09	02:10
20	5	5	9377	421	0:47	06:41
20	4	6	28340	813	1:45	06:49
20	6	7	64935	639	0:51	06:06
20	5	8	27736	473	2:33	07:34
21	5	4	74648	813	3:36	16:35
21	5	5	22378	862	2:03	5:09
21	5	6	164833	627	17:38	30:27
21	5	7	53297	345	10:50	16:59
21	3	8	228192	1318	24:13	39:59
22	5	4	19264	621	5:57	23:05
22	5	5	118535	640	14:58	26:38
22	5	6	91323	301	9:00	22:27
22	5	7	98931	497	7:32	18:12
22	2	8	105814	730	12:10	21:29
23	5	4	57254	1318	6:44	13:15
23	6	5	51970	306	6:03	11:01
23	5	6	32007	755	4:31	11:18
23	5	7	74648	354	1:21	04:17
23	5	8	40267	444	6:27	13:14
24	5	4	10596	302	1:39	06:19
24	5	5	13736	293	2:26	06:51
24	5	6	7354	140	1:12	02:51
24	5	7	27604	520	3:43	11:52
24	3	8	55772	307	11:14	15:42

Для частково-комбінаторних задач з дробово-лінійною функцією цілі та додатковими лінійними обмеженнями на розміщеннях була проведена серія розрахунків з 56 задач вимірностей 21 та 22. Як вихідні дані використовувалися задачі третьої серії експериментів, у яких k змінювалося в межах від 16 до 20 (до 21 для задач вимірності $m=22$). Результати розрахунків для $m=21$ наведено в таблиці 4.6.

Таблиця 4.6

**Результати числових експериментів для частково-комбінаторних
задач з дробово-лінійною функцією цілі**

№	№ задачі	k	P	τ	q	w	β	γ	T
1	1	16	4	14	143	130	86519	33223	4:47
2	2	20	4	10	90	71	276	26	0:02
3	3	18	4	10	89	76	148170	8225	11:42
4	4	20	5	21	219	201	1509	73	0:12
5	5	19	5	12	137	121	2234	311	0:12
6		20	5	12	156	134	7694	1449	0:37
7	6	19	5	13	134	114	61731	42696	3:28
8	7	17	5	7	59	40	3910	798	0:36
9		18	5	8	63	50	8167	754	0:33
10		19	5	11	110	94	10032	826	0:45
11		20	5	9	100	83	19134	912	1:48
12	8	17	5	13	117	103	27549	1632	1:43
13		18	5	8	80	65	26125	2189	1:51
14	9	16	6	10	84	71	2297	31	0:11
15	10	20	6	11	101	85	103739	821	10:21
16	11	16	6	11	105	92	50110	419	3:39
17		17	6	9	73	58	18081	109	1:18

Продовж. табл. 4.6

№	№ задачі	k	p	τ	q	w	β	γ	T
18		18	6	8	66	55	71497	170	5:04
19		19	6	11	109	96	145961	194	12:11
20	12	17	6	9	72	59	36074	1435	2:34
21		20	6	9	105	86	89815	1218	8:31
22	13	16	6	8	58	41	8232	484	0:30
23		19	6	11	97	81	144356	3767	12:02
24	14	16	7	12	95	77	5634	955	0:19
25		17	7	13	118	104	7783	750	0:30
26		18	7	15	125	114	13731	237	0:58
27		19	7	16	178	164	14492	171	1:18
28		20	7	19	222	207	30923	2	2:47
29		17	7	8	52	36	1470	435	0:05

Аналізуючи результати числових експериментів таблиці 4.6, можна зробити висновок, що зі збільшенням k в межах від 16 до 20 для кожної окремої задачі спостерігається тенденція до збільшення часу знаходження її розв'язку. При цьому час розв'язування задачі не перевищує час її розв'язування у випадку, коли $k = m$.

Приклади задач третьої серії розрахунків наведено в додатку В (приклади В.11, В.12). За умовами цих задач побудовано та розв'язано частково-комбінаторні задачі (див., наприклад, приклади В.13, В.14).

Аналізуючи результати досліджень, можна зробити висновок, що алгоритм розв'язування задачі (4.1)-(4.4) є практично ефективним у даній програмній реалізації для задач розглянутого класу при $m \leq 33$. Більша частина задач вимірностей $m > 33$ не розв'язується внаслідок обмеженості ресурсів мови програмування.

ВИСНОВКИ

Одним із сучасних і перспективних напрямів досліджень у сфері математичного програмування є евклідова комбінаторна оптимізація. Її актуальність зумовлена практичною значущістю комбінаторних множин для формулювання та розв'язання задач оптимізації, які описують економічні, соціальні та інші процеси.

У межах роботи розглянуто задачу теорії евклідової комбінаторної оптимізації на множині розміщень, що передбачає розробку алгоритмів її розв'язування. Проведено аналіз основних підходів у дискретній, зокрема комбінаторній, оптимізації. У ході дослідження встановлено, що евклідові комбінаторні множини, такі як множина розміщень, є ефективним інструментом для математичного моделювання задач оптимізації.

На основі аналізу властивостей комбінаторного многогранника сформульовано критерій оптимальності для евклідової комбінаторної задачі з дробово-лінійною цільовою функцією без додаткових обмежень. Цей критерій можна застосовувати для побудови аналітичних рішень у задачах на інших комбінаторних множинах.

У роботі запропоновано метод і алгоритм розв'язування умовних задач із дробово-лінійною цільовою функцією на множині розміщень. Вони є розвитком алгоритмів для розв'язування лінійних задач на розміщеннях. Виконано оцінки ефективності допоміжних алгоритмів, які є складовими основного методу, а також проведено числові експерименти. Результати експериментів підтвердили практичну ефективність запропонованого алгоритму.

Запропонований підхід придатний для розв'язання частково або повністю комбінаторних задач із дробово-лінійною цільовою функцією на множині розміщень, як умовних, так і безумовних.

Результати дослідження можуть бути використані в освітньому процесі Полтавського університету економіки і торгівлі на кафедрі комп'ютерних наук та інформаційних технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Сергієнко І. В.* Інформатика в Україні: становлення, розвиток, проблеми. – К.: Наукова думка, 1999. 354 с.
2. *Стоян Ю. Г., Ємець О. О.* Теорія і методи евклідової комбінаторної оптимізації. К.: Інститут системних досліджень освіти, 1993. 188 с.
3. *Ємець О. О., Колечкіна Л. М.* Задачі комбінаторної оптимізації з дробово-лінійними цільовими функціями: Монографія. К.: Наукова думка, 2005. 117 с.
4. *Стоян Ю. Г., Ємець О. О., Ємець Є. М.* Множини полірозміщень в комбінаторній оптимізації // Доповіді НАН України. 1999. № 8. С. 37-41.
5. *Козерацька Л. М.* Цілочислові задачі оптимізації: проблема стійкості і параметричний аналіз: Автореф. дис... д-ра фіз.-мат. наук: 01.05.01 / Інститут кібернетики ім. В.М.Глушкова НАН України. К., 1997. 32 с.
6. *Павлова Л. О.* Розробка та дослідження ПДС-алгоритмів для важкорозв'язуваних задач комбінаторної оптимізації: Автореф. дис... канд. фіз.-мат. наук: 01.05.01 / Інститут кібернетики ім. В. М. Глушкова НАН України. К., 1996. 19 с.
7. *Хіміч О. М.* Методи комп'ютерного дослідження математичних моделей з наближено заданими вихідними даними: Автореф. дис... д-ра фіз.-мат. наук: 01.05.02 / Інститут кібернетики ім. В. М. Глушкова НАН України. – К., 2003. 30 с.
8. *Кісельова О. М., Гарт Л. Л., Фірсов О. Д.* Генетичні алгоритми розв'язання оптимізаційних задач та їх застосування: Навч. посіб. Дніпропетр. нац. ун-т. – Дніпропетровськ, 2001. 48 с.
9. *Ємець О. О.* Теорія і методи комбінаторної оптимізації на евклідових множинах в геометричному проектуванні: Автореф. дис... д-ра фіз.-мат. наук: 01.05.01 / Ін-т кібернетики НАН України. К., 1997. 42 с.

- 10.Ємець О. О. Теорія і методи комбінаторної оптимізації на евклідових множинах в геометричному проектуванні: Дис. ... д-ра фіз.-мат. наук: 01.05.01. К., 1997. 242 с.
- 11.Ємець О. О., Ємець Є. М. Відсікання в лінійних частково комбінаторних задачах евклідової комбінаторної оптимізації // Доповіді НАН України. 2000. № 9. С. 105-109.
- 12.Ємець О. О., Барболіна Т. М., Черненко О. О. Розв'язування умовних задач з дробово-лінійною функцією цілі на множині розміщень // Доповіді НАН України. 2006. №11. С. 15-18.
- 13.Колєчкіна Л. М. Властивості задач комбінаторної оптимізації з дробово-лінійними цільовими функціями. Методи та алгоритми їх розв'язання: Дис. ... канд. фіз.-мат. наук: 01.05.02. Харків, 2002. 170 с.
- 14.Ємець Є. М. Дослідження властивостей математичних моделей комбінаторних задач оптимізації на полірозміщеннях та розробка методу і алгоритму комбінаторного відсікання: Дис. ... канд. фіз.-мат. наук: 01.05.02. Харків, 2002. 155 с.
- 15.Роскладка О. В. Задачі оптимізації на полікомбінаторних множинах: властивості та розв'язування: Дис. ... канд. фіз.-мат. наук: 01.05.01 / Інститут кібернетики ім. В. М. Глушкова НАН України. К., 2005. 163 с.
- 16.Барболіна Т. М. Методи й алгоритми розв'язування оптимізаційних задач на розміщеннях з додатковими умовами: Дис. ... канд. фіз.-мат. наук: 01.05.01 / Інститут кібернетики ім. В. М. Глушкова НАН України. К., 2005. 174 с.
- 17.Шило В. П. Методи розв'язання складних задач дискретної оптимізації: Автореф. дис... д-ра фіз.-мат. наук: 01.05.01 / Інститут кібернетики ім. В.М.Глушкова НАН України. К., 2003. 33 с.
- 18.Гребеннік І. В. Математичні моделі та методи комбінаторної оптимізації в геометричному проектуванні: Автореф. дис... д-ра техн. наук: 01.05.02 / Інститут проблем машинобудування НАН України. Харків, 2006. 34 с.

19. *Dantzig G. B.* Discrete-variable extremum problems // *Oper. Res.* 1957. Vol. 5, № 2. P. 266-277.
20. *Gomory R. E.* An algorithm for integer solutions to linear programs. *Recent Advances Math. Program*, McGraw-Hill Book Co., Inc., 1963, P. 269-302.
21. *Dalton R. E., Llewellyn R. W.* An extension of the Gomory mixed-integer algorithm to mixed-discrete variables, *Manag. Sci.*, 1966. № 7, P. 562-575.
22. *Ємець О. О., Роскладка О. В.* Задачі оптимізації на полікомбінаторних множинах: властивості та розв'язування: Монографія. Полтава: РВЦ ПУСКУ, 2006. 144 с.
23. *Liudmyla Koliechkina, Olena Dvirna* Using Models of Combinatorial Optimization Problem to Estimate the Parameters of an Intelligent System. 4th International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2024), 25-27.09.2024. Cambridge, MA, USA, pp. 385-391.
24. *Черненко О.О., Гаркуша С.В., Кошова О.П., Стусь С.* Аналіз алгоритмів розв'язування умовної задачі оптимізації дробово-лінійної цільової функції на множині розміщень // *Вісник КрНУ імені Михайла Остроградського*. Випуск 3/2024 (146), С. 106-116.
25. *Sergiy Yakovlev, Oksana Pichugina, Liudmyla Koliechkina.* Combinatorial point configurations and polytopes. Łódź University Press, 2023, 232p., monografia DOI: <https://doi.org/10.18778/8331-391-72022>
26. *Pichugina, L. Koliechkina and T. Chilikina,* Multicriteria Combinatorial Optimization Model of an Infocommunication System, 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 13-16, doi: 10.1109/PICST54195.2021.9772124.