

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Український державний хіміко-технологічний університет»

На правах рукопису

ПЕРЕТЯТЬКО АНАСТАСІЯ СЕРГІЇВНА



УДК 519.85

НАПІВВИЗНАЧЕНА ОПТИМІЗАЦІЯ ДЛЯ РОЗВ'ЯЗУВАННЯ
ЗАГАЛЬНИХ КВАДРАТИЧНИХ ЗАДАЧ

01.05.02 – математичне моделювання
та обчислювальні методи

Дисертація на здобуття наукового ступеня
кандидата фізико-математичних наук

Науковий керівник
Косолап Анатолій Іванович,
доктор фізико-математичних наук,
професор

*Цей примірник дисертаційної роботи
ідентичний за змістом з іншими,
поданими до спеціалізованої вченої
ради Д 64.052.02*

Вчений секретар спецради Д 64.052.02



Л.В. Колесник

Дніпропетровськ – 2015

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
1 НАПІВВИЗНАЧЕНА ОПТИМІЗАЦІЯ	14
1.1 Постановка задачі напіввизначеної оптимізації	14
1.1.1 Пряма задача напіввизначеної оптимізації (SDP).....	14
1.1.2 Властивості (напів)визначених матриць	16
1.1.3 Додатно напіввизначений конус	17
1.1.4 Двоїста задача SDP	18
1.2 Методи внутрішньої точки для розв'язування задач SDP	20
1.3 Інші методи для розв'язування задач напіввизначеної оптимізації	26
1.4 Застосування SDP	27
1.4.1 Загальні квадратичні задачі.....	27
1.4.2 Напіввизначена релаксація	28
1.4.3 Максимум норми вектора на перетині еліпсоїдів.....	30
1.4.4 Задачі булевої оптимізації.....	33
1.5 Огляд програмного забезпечення для розв'язування задач SDP.....	34
1.6 Симплекс-метод для задач лінійної оптимізації.....	38
1.7 Методи знаходження власних значень матриці та відповідних їм власних векторів	41
2 НАПІВВИЗНАЧЕНИЙ СИМПЛЕКС-МЕТОД ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ НАПІВВИЗНАЧЕНОЇ ОПТИМІЗАЦІЇ	44
2.1 Метод спряжених напрямів для визначення додатної напіввизначеності симетричної матриці	44
2.1.1 Постановка задачі і метод розв'язування	45
2.1.2 Алгоритм методу спряжених напрямів	52
2.1.3 Приклад	52
2.1.4 Порівняльні чисельні експерименти	55
2.2 Напіввизначений симплекс-метод.....	56

	3
2.2.1 Перетворення задачі SDP до задачі лінійного програмування	58
2.2.2 Пошук коефіцієнта цільової функції	62
2.2.3 Розв'язування квадратичної підзадачі	66
2.2.4 Алгоритм напіввизначеного симплекс-методу	67
2.2.5 Приклад	68
2.2.6 Переваги напіввизначеного симплекс-методу	69
2.3 Збіжність напіввизначеного симплекс-методу для напіввизначеної оптимізації	71
2.4 Чисельні експерименти з напіввизначеним симплекс-методом	73
2.5 Висновки розділу	76
3 SDP ДЛЯ ЗАГАЛЬНИХ КВАДРАТИЧНИХ ЗАДАЧ	77
3.1 Розв'язування загальних квадратичних задач	77
3.1.1 Постановка задачі і метод розв'язування	77
3.1.2 Порівняльні чисельні експерименти	79
3.2 Уточнення напіввизначеної релаксації	87
3.2.1 Розв'язування задачі максимізації норми вектора на перетині еліпсоїдів	87
3.2.2 Порівняльні чисельні експерименти	93
3.2.3 Задачі, для яких напіввизначена релаксація є точною	100
3.3 Розв'язування квадратичних задач з булевими змінними	106
3.3.1 Постановка задачі і метод розв'язування	106
3.3.2 Порівняльні чисельні експерименти	109
3.4 Переваги напіввизначеної релаксації	110
3.5 Висновки розділу	111
4 ПРИКЛАДНІ ЗАДАЧІ НАПІВВИЗНАЧЕНОЇ ОПТИМІЗАЦІЇ	113
4.1 Модель кластеризації даних	113
4.1.1 Постановка задачі і метод розв'язування	113
4.1.2 Порівняльні чисельні експерименти	118
4.2 Модель розміщення сенсорних датчиків у мережі	121
4.2.1 Постановка задачі і метод розв'язування	121

	4
4.2.2 Порівняльні чисельні експерименти	126
4.3 Задача Max-Cut	133
4.3.1 Постановка задачі і метод розв'язування	133
4.4 Програмне забезпечення для розв'язування задач напіввизначеної оптимізації	135
4.5 Висновки розділу	139
ВИСНОВКИ	140
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	143
ДОДАТКИ	163
Додаток А Приклади, для яких метод внутрішньої точки не може знайти розв'язок	163
Додаток Б Вхідні дані для задач.....	177
Додаток В Акт впровадження	181

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SDP – напіввизначена оптимізація (пряма задача напіввизначеної оптимізації);

SDD – двоїста задача напіввизначеної оптимізації;

$C \bullet X = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$ – скалярний добуток двох матриць;

$X \succeq 0$ – матриця X є додатно напіввизначеною;

$X \succ 0$ – матриця X є додатно визначеною;

$\text{rank } X$ – ранг матриці X ;

S^n – сукупність симетричних матриць ($n \times n$);

S_+^n – сукупність симетричних додатно напіввизначених матриць;

S_{++}^n – сукупність симетричних додатно визначених матриць;

R^n – евклідовий простір;

$\|x\|$ – евклідова норма вектора в евклідовому просторі;

QCQP – загальна задача квадратичної оптимізації;

Max-cut – задача пошуку максимального розрізу графа.

ВСТУП

Актуальність теми. Оптимізаційні моделі складних систем в галузях ринкової економіки, фінансів, планування, управління проектами, штучному інтелекті, оптимальному проектуванні, хімічних технологіях, комп'ютерних системах призводять до необхідності розв'язування задач, які можна представити у вигляді загальних квадратичних задач, тобто задач, в яких необхідно мінімізувати квадратичну цільову функцію при квадратичних обмеженнях (функції обмежень та цільова функція є неперервними та гладкими, але можуть бути неопуклими). Задача скінченновимірної мінімізації квадратичної функції при квадратичних обмеженнях має ефективний розв'язок, якщо цільова функція і обмеження – опуклі. Якщо ж принаймні одна з функцій не є опуклою, то така задача стає достатньо складною і в загальному випадку багатоекстремальною. Наразі для розв'язування таких задач не існує поліноміальних методів. Методи гілок та границь потребують експоненціального часу для знаходження змінних моделей і тому можуть бути використані тільки для розв'язування задач невеликої розмірності. Методи випадкового пошуку іноді дозволяють знайти оптимальні розв'язки, але перевірка цієї оптимальності потребує експоненціального часу. Для розв'язування складних тестових задач з відомим розв'язком ці методи інколи знаходять оптимальну точку, але в багатьох випадках ці розв'язки далекі від оптимальних.

Прогрес в математичному моделюванні складних систем можливий тільки з запровадженням нових методів, які дозволять ефективно розв'язувати класи складних оптимізаційних задач. Пошуку таких методів присвячена велика кількість публікацій. Значний вклад в математичне моделювання та обчислювальні методи внесли вчені Н. З. Шор [147, 169-171], Б. Н. Пшеничний [158, 159], І. В. Сергієнко [160, 161], Ю. Г. Стоян [163], О. О. Ємець [163], В. І. Норкін [116, 148-149], О. М. Литвин [161], С. І. Ляшко [145], А. І. Косолап [123-125], В. Ф. Дем'янов [115] та інші.

Декілька останніх десятиліть пильну увагу приділяють оптимізаційним

задачам, в яких змінні є не вектором, а симетричною матрицею, яка до того ж є додатно напіввизначеною. У англomовній літературі термін «напіввизначене програмування» перекладається як «Semidefinite Programming» (SDP), рідше – «Semidefinite Optimization» (SDO). Значний внесок в напіввизначене програмування зробили L. Vandenberghe [99], S. P. Boyd [60, 99, 103], Y. E. Nesterov [71], A. S. Nemirovskii [71], F. Alizadeh [3, 4, 84], M. J. Todd [96, 97], C. Helmberg [5, 40-43], B. Borchers [16, 17], Y. Ye [10, 13, 64, 66, 92, 107], а також Н. З. Шор [169, 170], А. І. Косолап [51, 122-142, 152, 155], В. Г. Жадан [117, 118].

В задачах SDP мінімізується лінійна функція $C \bullet X$ з лінійними обмеженнями $A_i \bullet X = b_i$, в якій змінною є додатно напіввизначена матриця X [99]. Оскільки обмеження цієї задачі також включають нелінійну умову додатної напіввизначеності матриці, то допустима множина для задачі SDP є нелінійною, але опуклою, тому задачі напіввизначеного програмування належать до класу опуклих задач оптимізації.

Уперше задача SDP була сформульована у 1963 році [9]. Підвищення активності досліджень у дев'яностих роках двадцятого століття було мотивовано відкриттям нових прикладних задач напіввизначеної оптимізації в окремих областях у поєднанні з розробкою нових методів. Не так давно SDP почали успішно застосовувати для розв'язування фінансових задач [37]. Проводяться спроби застосування SDP в розробці систем розпізнавання мови, в алгоритмах калібрування фотокамер [1, 2], в задачі локалізації сенсорних датчиків у мережі [38, 53, 62, 66, 86, 105]. Застосування SDP в комбінаторній оптимізації сфокусовано на розв'язуванні задач NP-складності: задачі max-cut [3, 5, 31, 35, 43, 54, 56, 57, 74, 75, 88, 90, 96, 107, 110], задачі розфарбування графа мінімальною кількістю кольорів [43, 46, 56, 57] та булевого програмування [3, 43, 56, 110]. Задачі SDP часто виникають в теорії керування [42, 60, 93, 96, 99]. Велика кількість геометричних задач, що включають квадратичні функції, можуть бути представлені як задачі напіввизначеного програмування [31, 35, 57, 96, 99, 107]. Існує багато типів задач на власні числа,

які можуть бути сформульовані як задачі SDP [3, 5, 35, 42, 43, 57, 75, 79, 96, 99]. Відносно нова галузь в оптимізації – кододатне програмування, яку можна розглядати як узагальнення напіввизначеного програмування, також почала розвиватися з винаходом ефективних методів для розв’язування задач напіввизначеної оптимізації [15, 18, 28, 44, 78]. Прогрес в цій області, проте, є дещо повільним через той факт, що на практиці теоретично ефективні методи, розроблені для SDP, є фактично неефективними через затрати пам’яті та часу, особливо для напіввизначеної релаксації задач булевої оптимізації.

Розглянуті вище прикладні задачі формулюються як задачі загальної квадратичної оптимізації, які шляхом напіввизначеної релаксації перетворюються до задач SDP (під релаксацією будемо розуміти послаблення обмежень початкової задачі). Основна цінність задач SDP полягає в тому, що вони дозволяють розв’язувати багатоекстремальні квадратичні задачі за допомогою опуклої напіввизначеної релаксації [19, 27, 64, 86], яка використовує перетворення квадратичного виразу $x^T A x$ до скалярного добутку матриць $A \bullet x x^T$ або $A \bullet X$, де X – додатно напіввизначена матриця рангу одиниця. Застосування напіввизначеної релаксації дозволяє знаходити нижні оцінки цільової функції в загальних квадратичних задачах або точку глобального мінімуму з заданою точністю. Причому, на відміну від інших методів, перевірка оптимальності знайденого розв’язку не потребує експоненціального часу та є досить простою (достатньо перевірити ранг матриці X). Таким чином, напіввизначена релаксація значно розширює сферу використання напіввизначеної оптимізації. Її альтернативою при розв’язанні загальних квадратичних задач є двоїстий метод, запропонований Н. З. Шором, але двоїста задача є негладкою та досить складною.

На даний момент існують декілька типів методів для розв’язування задач SDP. Більшість методів для розв’язування задач напіввизначеного програмування базується на методах, розроблених для розв’язування задач лінійного програмування. Перший метод, запропонований у 1988 році для розв’язування задач SDP, – метод еліпсоїдів (був узагальненням аналогічного

методу для лінійного програмування [167, 171]) – хоча і зводив задачу до поліноміального класу, проте на практиці виявився повільним [99]. Більшість методів внутрішньої точки для лінійного програмування також узагальнені для задач напіввизначеної оптимізації. Ці методи мають поліноміальну складність і досить добре реалізуються на практиці. Методи внутрішньої точки для задач малої та середньої розмірності виявилися досить ефективними та отримували оптимальні розв'язки з заданою точністю за короткі проміжки часу. Їх продуктивність на задачах великої розмірності, проте, не була вражаючою [20].

Тобто, на даний момент основні дослідження ведуться в методах внутрішньої точки, але й ці методи мають недоліки. Вони значно збільшують розмірність початкової задачі, обмежені використанням тільки для задач з нульовим розривом двоїстості, вимагають лінійну незалежність обмежень. Існують й інші методи розв'язування задачі SDP, але вони не отримали широкого розповсюдження [36, 40, 52, 58, 59, 63, 69, 70, 80, 81, 83, 99, 111, 117, 118]. В методах SDP потрібно враховувати і те, що необхідно весь час контролювати додатну напіввизначеність матриць, а це досить трудомісткий процес. Перевірка додатної напіввизначеності можлива тільки алгоритмічно, що ускладнює розробку ефективних методів напіввизначеної оптимізації.

Враховуючи практичну ефективність симплекс-методу для задач лінійного програмування та його нечутливість до початкової точки, можливість просто встановлювати пустоту та необмеженість допустимої області, були прикладені значні зусилля для його узагальнення для задач SDP, але тільки версія такого узагальнення, яка розглядається в даній роботі, виявилась ефективною.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась у відповідності з планами наукових досліджень Державного вищого навчального закладу «Український державний хіміко-технологічний університет» в рамках наукової теми «Дослідження шляхів використання інформаційних та комп'ютерних технологій при вирішенні регіональних проблем» (№ держреєстрації 0112U004341, 2012-2014 рр.), де дисертант була автором розділів «3.8 Задача кластеризації даних»,

«3.9 Задача розміщення сенсорних датчиків у мережі», в яких використані результати дисертації для розв'язання прикладних задач.

Мета і завдання дослідження.

Мета роботи – обґрунтування збіжності та вдосконалення напіввизначеного симплекс-методу для розв'язування задач напіввизначеної оптимізації, застосування та удосконалення напіввизначеної релаксації для розв'язування прикладних задач квадратичної оптимізації (задач з булевими змінними, задачі кластеризації даних, задачі розміщення сенсорних датчиків у мережі, задачі пошуку максимального розрізу графа (max-cut)).

Для досягнення поставленої мети потрібно розв'язати такі *завдання*:

– удосконалити напіввизначений симплекс-метод для розв'язування задач напіввизначеної оптимізації, що ґрунтується на симплекс-методі для розв'язування задач лінійної оптимізації, дослідити його збіжність та засоби її прискорення, визначити його теоретичні та практичні переваги;

– удосконалити напіввизначену релаксацію для розв'язування загальних квадратичних задач і прикладних задач, які можуть бути описані квадратичними оптимізаційними моделями, для отримання кращих оцінок цільової функції;

– розглянути сфери застосування напіввизначеної оптимізації;

– виконати чисельні експерименти для перевірки ефективності напіввизначеної релаксації для оптимізаційних моделей різних класів прикладних задач квадратичної оптимізації.

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

Предмет дослідження – напіввизначена релаксація загальних квадратичних задач і застосування напіввизначеного симплекс-методу для розв'язування релаксованих задач.

Методи дослідження. При розв'язуванні поставлених задач використовувалася багатовимірна евклідова геометрія, опуклий аналіз, математичне моделювання, теорія та чисельні методи оптимізації. Для

розв'язування загальних квадратичних задач використовувався метод напіввизначеної релаксації, точна квадратична регуляризація, напіввизначений симплекс-метод та прямо-двоїстий метод внутрішньої точки.

Наукова новизна одержаних результатів полягає в обґрунтуванні та удосконаленні напіввизначеного симплекс-методу як альтернативи існуючим прямо-двоїстим методам внутрішньої точки для розв'язування задач напіввизначеної оптимізації. Це розширило межі ефективного використання математичного моделювання складних систем, зокрема тих систем, які описуються загальними квадратичними функціями. Зокрема, в дисертації:

- *удосконалено* процедуру оберненої ітерації, яка використовується для визначення додатної напіввизначеності матриці, шляхом використання методу спряжених напрямів, що за результатами чисельних експериментів дозволило підвищити точність розрахунків і прискорити збіжність до власного вектора матриці;

- *уперше строго доведено* збіжність напіввизначеного симплекс-методу, який вдосконалено процедурою оберненої ітерації з використанням спряжених напрямів, встановлено його теоретичні та чисельні переваги;

- *удосконалено* напіввизначену релаксацію шляхом використання точної квадратичної регуляризації та інших перетворень для загальних квадратичних задач і широкого кола прикладних задач (задачі розміщення сенсорних датчиків у мережі, задачі кластеризації даних, квадратичних задач з булевими змінними, задачі *max-cut*), що на відміну від існуючих методів дозволило знаходити кращі розв'язки початкової задачі;

- *доведено точність* напіввизначеної релаксації для окремих класів задач максимізації евклідової норми вектора на опуклій множині;

- *уперше запропоновано* та перевірено на практиці нову процедуру знаходження верхніх і нижніх оцінок цільової функції, яка дозволяє отримувати оптимальні розв'язки у загальних задачах квадратичної оптимізації.

Практичне значення одержаних результатів. Удосконалений напіввизначений симплекс-метод може бути використаний для розв'язування

широкого кола практичних задач. Зокрема, задач кластеризації даних, розміщення датчиків у мережі, в алгоритмах калібрування фотокамер, для розфарбування графів та інших. Розроблене програмне забезпечення для розв'язування задач напіввизначеної оптимізації реалізує два найбільш популярних та ефективних методи (прямо-двоїстий метод внутрішньої точки та напіввизначений симплекс-метод), ефективність яких перевірена за допомогою чисельних експериментів над задачами різної розмірності. Практичне значення отриманих результатів підтверджується актом про впровадження результатів дисертаційної роботи у навчальний процес Державного вищого навчального закладу «Український державний хіміко-технологічний університет» (додаток В).

Особистий внесок здобувача. Всі результати дисертаційної роботи отримано автором особисто. У роботах, опублікованих у співавторстві, дисертанту належать такі результати: у [131] вперше використала метод спряжених напрямів у процедурі оберненої ітерації для визначення додатної напіввизначеності матриці; у [132, 135] уперше дала строге доведення збіжності напіввизначеного симплекс-методу; у [129, 155] удосконалила напіввизначений симплекс-метод шляхом використання методу спряжених напрямів у процедурі оберненої ітерації; у [132, 134] встановила теоретичні та чисельні переваги напіввизначеного симплекс-методу; у [126–128, 130, 133, 136–140, 142, 152] удосконалила напіввизначену релаксацію шляхом використання точної квадратичної регуляризації та інших перетворень; у [142] уперше довела точність напіввизначеної релаксації для окремих класів задач; у [51, 122, 128] уперше запропонувала та перевірила на практиці нову процедуру знаходження верхніх і нижніх оцінок цільової функції у загальних задачах квадратичної оптимізації.

Апробація результатів дисертації. Результати дисертаційної роботи пройшли апробацію на міжнародних і всеукраїнських конференціях «Математичне та програмне забезпечення інтелектуальних систем» (Дніпропетровськ, 2009 р.), «Dynamical system modelling and stability investigation» (Київ, 2011 р.), «Математичне та імітаційне моделювання систем»

(Чернігів, 2011-2013 рр.), «Обчислювальна та прикладна математика» (Київ, 2011 р., 2014 р.), «Інформатика та системні науки» (Полтава, 2012 р., 2014 р., 2015 р.), «Контроль і управління в складних системах» (Вінниця, 2012 р.), «Хімія та сучасні технології» (Дніпропетровськ, 2013 р.), «Современные проблемы математики и ее приложения в естественных науках и информационных технологиях» (Харків, 2013 р.), «Computer science & engineering» (Львів, 2013 р.), «Актуальные проблемы современной науки» (Уфа, 2013 р.).

Публікації. Основні результати дисертації опубліковано у 18 наукових роботах [51, 77, 122, 126–132, 135, 137, 142, 152–156], з них 8 статей (7 – це статті у наукових журналах та збірниках наукових праць, які входять до переліку фахових видань України з фізико-математичних наук ([122, 127–129, 131, 132, 137], у тому числі одна стаття у Міжнародному науково-технічному журналі «Проблемы управления и информатики», який введений в базу даних Scopus [132]; 1 стаття в іноземному періодичному виданні [77]), 10 – це роботи у матеріалах наукових конференцій ([51, 126, 130, 135, 142, 152–156]).

Дисертаційна робота виконана під керівництвом доктора фізико-математичних наук, професора Косолапа А.І.

Автор вдячний науковому керівнику за корисні поради та зауваження у процесі підготовки дисертації.

РОЗДІЛ 1

НАПІВВИЗНАЧЕНА ОПТИМІЗАЦІЯ

У цьому розділі наводяться основні відомості теорії та сучасних методів напіввизначеної оптимізації із зазначенням їх переваг та недоліків. Розглядається застосування напіввизначеної релаксації для розв'язування різних класів задач квадратичної оптимізації. Здійснюється огляд літературних джерел, пов'язаних з темою дисертації, та існуючого програмного забезпечення для розв'язування задач напіввизначеної оптимізації та її прикладних задач. Наводяться основні положення симплекс-методу для задач лінійного програмування, які використано у напіввизначеному симплекс-методі, та методи знаходження власних значень матриці та відповідних їм власних векторів.

1.1 Постановка задачі напіввизначеної оптимізації

1.1.1 Пряма задача напіввизначеної оптимізації (SDP)

Пряма задача SDP розглядає проблему мінімізації лінійної функції $C \bullet X$:

$$\min \{ C \bullet X \mid A_i \bullet X = b_i, \quad i = 1, \dots, m, \quad X \succeq 0 \}, \quad (1.1)$$

де C – симетрична матриця ($n \times n$);

A_i – симетричні матриці ($n \times n$);

b – вектор-стовпчик розмірності m , де b_1, \dots, b_m – компоненти вектора b ;

X – симетрична додатно напіввизначена матриця ($n \times n$).

Позначення $C \bullet X$ означає скалярний добуток двох матриць [35]:

$$C \bullet X = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}.$$

Таке позначення скалярного добутку матриць є загальноприйнятим в напіввизначеній оптимізації.

Задача (1.1) називається також задачею *напіввизначеного програмування*. В цій задачі необхідно знайти додатно напіввизначену матрицю X , що задана умовами (1.1).

Слід зазначити, що обмеження в задачі SDP є нелінійними, оскільки допустима область визначається замкнутим опуклим небагатогранним конусом додатно напіввизначених матриць. Границя цього конуса є множиною додатно напіввизначених матриць (див. п. 1.1.3). Таким чином, загальні характеристики допустимої області SDP наступні: вона опукла, так як складається з перетину гіперплощин та опуклого конусу напіввизначених матриць. Детальніше структура допустимої області задачі SDP приведена С. Helmberg у [43] та Dattorro у [26].

Розглядається також узагальнення задачі (1.1), в якій невідомими є послідовність додатно напіввизначених матриць. Ця задача зводиться до вигляду:

$$\min \{ C \bullet X \mid A_i \bullet X_j = b_i, X_1 \succeq 0, \dots, X_m \succeq 0, j=1, \dots, m \}. \quad (1.2)$$

Тобто будь-яку задачу SDP з будь-якою кількістю змінних з різними розмірностями можна сформулювати як задачу SDP з однією додатно напіввизначеною змінною:

$$X_1 \succeq 0, X_2 \succeq 0, \dots, X_m \succeq 0 \Leftrightarrow \begin{pmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & X_m \end{pmatrix}. \quad (1.3)$$

Відповідно блочно-діагональними будуть матриці обмежень.

Якщо допустити, що $X_j \succeq 0 \forall j$ – матриці розмірності 1×1 , то з (1.3) можна зробити висновок, що задача (1.2) є задачею лінійного програмування.

Області задач лінійної та нелінійної оптимізації можна представити наступною схемою (рис.1.1). З рис. 1.1 можна побачити, що задачі

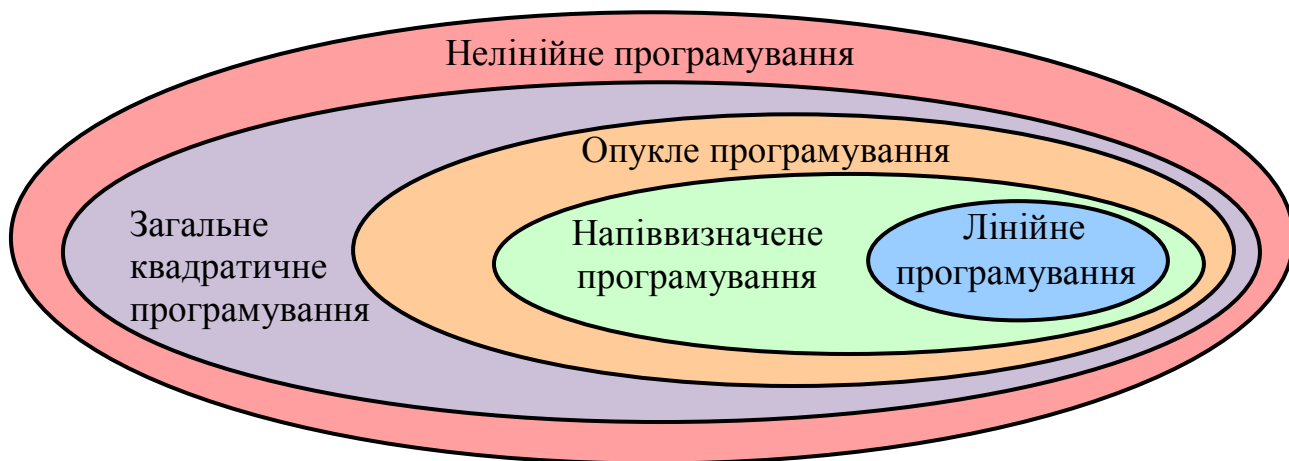


Рисунок 1.1 – Зв’язок лінійного та напіввизначеного програмування

напіввизначеного програмування є більш загальними в порівнянні з задачами лінійного програмування, але, як і лінійне програмування, задачі SDP відносяться до опуклих задач оптимізації. Саме тому розробка методів напіввизначеної оптимізації йшла по шляху удосконалення методів лінійного програмування, і багато методів для розв’язування задач лінійного програмування мають узагальнення для задач напіввизначеного програмування. Але це узагальнення не є простим [99], так як в задачах SDP можливий розрив двоїстості та ці задачі є нелінійними.

1.1.2 Властивості (напів)визначених матриць

Перелічимо деякі важливі властивості симетричних додатно напіввизначених (визначених) матриць, які будуть використані в даній роботі.

Нехай X – симетрична матриця ($n \times n$), тоді X називається *додатно напіввизначеною (psd) матрицею*, якщо $v^T X v \geq 0$ для будь-яких $v \in R^n$ [168].

Нехай X – симетрична матриця ($n \times n$), тоді X називається *додатно визначеною (pd) матрицею*, якщо $v^T X v > 0$ для будь-яких $v \in R^n$, $v \neq 0$ [168].

Нехай X та Y – будь-які симетричні матриці. Запис “ $X \succeq 0$ ” використовується для позначення того, що матриця X є симетричною та напіввизначеною. Запис “ $X \succeq Y$ ” означає, що $X - Y \succeq 0$. Запис “ $X \succ 0$ ” означає, що матриця X є симетричною та додатно визначеною.

Усі власні значення додатно напіввизначеної матриці є більшими або

дорівнюють нулю. Найменше власне значення додатно напіввизначеної матриці дорівнює нулю [56].

Якщо $X \succeq 0$, тоді визначники всіх головних мінорів матриці X є невід'ємними дійсними числами (критерій Сильвестра) [56]. Для симетричної матриці розмірності $n \times n$ число таких визначників буде дорівнювати $2^n - 1$.

Будь-яка додатно напіввизначена матриця A рангу n може бути записана у вигляді суми матриць рангу одиниця $A = \sum_{i=1}^n \alpha_i Z_i \succeq 0$, де $Z_i = z_i z_i^T$ – матриці рангу одиниця, z_i – вектор-стовпчик, α_i – скаляр, $\alpha_i \geq 0 \forall i$. [168 с. 542].

Якщо $X \succeq 0$, то тоді існує таке Y , що $X = YY^T$ та виконується умова $\text{rank } X = \text{rank } Y$ [56].

Таким чином, умова $X \succeq 0$ в задачі напіввизначеної оптимізації не може бути записана аналітично у вигляді простої формули, що породжує проблеми при розробці методів розв'язування задачі (1.1).

1.1.3 Додатно напіввизначений конус

Позначимо через S^n сукупність симетричних матриць ($n \times n$), через S_+^n – сукупність симетричних додатно напіввизначених матриць, через S_{++}^n – сукупність симетричних додатно визначених матриць (див. п. 1.1.2).

Відомо, що множина $S_+^n = \{X \in S^n \mid X \succeq 0\}$ є замкнутим опуклим небагатограним конусом [35].

Матриця A є внутрішньою точкою додатно напіввизначеного конуса, якщо:

- $A \succ 0$ – тобто матриця A є симетричною та додатно визначеною;
- $A \in S_+^n$ та $\text{rank } A = n$, де n – розмірність матриці A [26].

Матриця A належить границі додатно напіввизначеного конуса, якщо:

– $A \succeq 0$, $A \neq 0$ – тобто матриця A є симетричною та додатно напіввизначеною, але не додатно визначеною;

- $A \in S^n$ та $\min\{\lambda_i(A), i = 1, \dots, n\} = 0$ – тобто матриця A має хоча б одне

нульове власне число ($\lambda_i(A)$ – множина власних чисел матриці A);

– $A \in S_+^n$ та $\text{rank } A < n$, де n – розмірність матриці A [26].

Особливої уваги заслуговують матриці з S_+^n , що мають ранг 1, – вони є твірними (крайніми променями) конусу додатно напіввизначених матриць – це означає, що їх лінійні комбінації задають усі інші матриці (рангу більше 1) додатно напіввизначеного конусу [141].

На рис. 1.2 зображено конус додатно напіввизначених матриць S_+^2 , лінеаризований до евклідового простору R^3 за допомогою оператора svec [26].

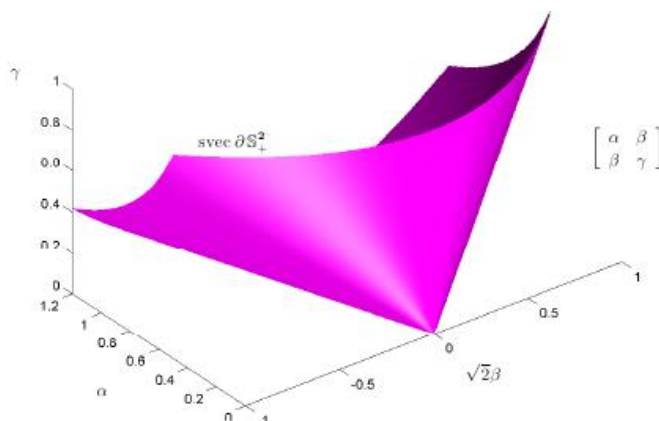


Рисунок 1.2 – Конус додатно напіввизначених матриць розмірності 2

Складність конусу додатно напіввизначених матриць полягає в тому, що множина крайніх променів є нескінченною, але кожні два з цих променів є сусідніми [56], тобто їх опукла комбінація є завжди крайнім променем S_+^n .

1.1.4 Двоїста задача SDP

Двоїста задача напіввизначеного програмування (SDD) до задачі SDP (1.1) визначається наступним чином:

$$\max \left\{ b^T y \mid \sum_{i=1}^m A_i^T y_i + Z = C, Z \succeq 0 \right\}, \quad (1.4)$$

де C – симетрична матриця ($n \times n$);

A_i – симетричні матриці ($n \times n$);

b – вектор-стовпчик розмірності m ;

y – вектор-стовпчик розмірності m , де y_1, \dots, y_m – двоїсті змінні – компоненти вектора y ;

Z – симетрична додатно напіввизначена матриця (вільна змінна).

В задачі (1.4) необхідно знайти y та Z .

Теорема 1.1. [123] *Якщо X – допустимий розв’язок задачі SDP та (y, Z) – допустимий розв’язок задачі SDD, то зазор двоїстості (різниця між цільовими функціями прямої та двоїстої задач) дорівнює $C \bullet X - \sum_{i=1}^m y_i b_i = Z \bullet X \geq 0$.*

Якщо $C \bullet X - \sum_{i=1}^m y_i b_i = 0$, тоді X та (y, Z) будуть оптимальними розв’язками задач SDP та SDD відповідно, і, до того ж, зазор двоїстості $Z \bullet X = 0$.

При певних умовах зазор двоїстості дорівнює нулю. Одна з таких умов сформульована в наступній теоремі. Позначимо через p^* – значення цільової функції прямої задачі (1.1), через d^* – значення цільової функції двоїстої задачі (1.4).

Теорема 1.2 (умова Слейтера). [123] *Якщо існує строго допустимий розв’язок задачі (1.1) (матриця X додатно визначена) і p^* скінченне, то $p^* = d^*$. Вірно і зворотне твердження: якщо розв’язок задачі (1.4) строго допустимий (різниця матриць в обмеженнях додатно визначена) та d^* скінченне, то $d^* = p^*$.*

Теорема 1.3 (умова Каруша-Куна-Таккера (ККТ)). [123] *Змінні X, y, Z є оптимальним розв’язком прямої та двоїстої задач, якщо і тільки якщо виконуються наступні умови:*

$$\sum_{i=1}^m A_i^T y_i + Z - C = 0,$$

$$b_i - A_i \bullet X = 0,$$

$$ZX = 0,$$

$$Z \succeq 0, X \succeq 0.$$

Умови *ККТ* використовуються для побудови прямо-двоїстих методів внутрішньої точки.

Двоїста теорія у напіввизначеній оптимізації є слабшою, ніж двоїста теорія у лінійному програмуванні. Для задачі *SDP* можливий розрив двоїстості навіть у тому випадку, коли пряма та двоїста задачі мають скінченний розв'язок. Тому використання двоїстості у методах для розв'язку задач *SDP* є обмеженим.

1.2 Методи внутрішньої точки для розв'язування задач *SDP*

Розглянемо найпоширеніші на даний момент методи для розв'язування задач напіввизначеної оптимізації – прямо-двоїсті методи внутрішньої точки.

Методи внутрішньої точки у математичному програмуванні є обширною та найдинамічнішою галуззю досліджень після розробки симплекс-методу для лінійного програмування. Показано, що для задач великої розмірності метод внутрішньої точки для задачі лінійного програмування по швидкості збіжності може перевершувати симплекс-метод [99].

Більшість методів для розв'язування задач напіввизначеного програмування базуються на методах, розроблених для розв'язування задач лінійного програмування. Нестеров і Неміровський узагальнили у [71] метод внутрішньої точки лінійного програмування для розв'язування опуклих задач оптимізації. Вони показали, що задачі лінійного програмування, опуклого квадратичного програмування з опуклими квадратичними обмеженнями та напіввизначеного програмування можуть бути розв'язані за „поліноміальний час”. Алізаде у [3] показав, що багато відомих методів внутрішньої точки для задач лінійного програмування можуть бути трансформовані у методи для напіввизначеного програмування. З того часу приділяється значна увага вдосконаленню методів внутрішньої точки для розв'язування задач *SDP* [4, 5, 71]. Саме створення поліноміального методу внутрішньої точки для задач *SDP* в кінці 80-х років стало головною причиною зростання інтересу до задач *SDP*.

Методи внутрішньої точки мають багато варіацій [3, 5, 8, 12, 21, 23, 31, 34, 35, 37, 41, 56, 58, 61, 67, 70, 71, 82, 85, 88, 96, 99, 102, 104, 106-108, 118, 123]. Головний недолік методів внутрішньої точки для напіввизначеної оптимізації –

їх незастосовність до задач великої розмірності, на відміну від задач лінійного програмування. Це пов'язано з великою кількістю рівнянь, що визначають оптимальний розв'язок прямої та двоїстої задач. Для типової робочої станції розмірність задачі SDP при розв'язуванні її методом внутрішньої точки обмежується 500, а кількість обмежень – 2000 [40]. У зв'язку з цим пошуки ефективніших методів тривають.

Методи внутрішньої точки реалізовані у багатьох пакетах: CSDP, DSDP, SDPT3, SDPA, SEDUMI (див. п. 1.5).

Головна ідея різних модифікацій методів внутрішньої точки полягає в тому, щоб позбутися „важкого” обмеження $X \succeq 0$ за допомогою додавання до цільової функції так званої „бар'єрної функції”.

Розглянемо задачу напіввизначеної оптимізації в канонічному вигляді (1.1) та двоїсту до неї (1.4). Розглянемо прямо-двоїстий метод внутрішньої точки [23] для розв'язування задач (1.1) та (1.4). Першим припущенням методу є існування допустимого розв'язку задачі (1.1), який задовольняє умові Слейтера (теорема 1.2). Другим припущенням методу є лінійна незалежність матриць A_i , $i = 1, \dots, m$ (тобто будь-яка матриця A_i не може бути виражена через лінійну комбінацію інших матриць A_i).

Згідно з теоремою 1.3, необхідні умови екстремума для задач (1.1) та (1.4) мають вигляд [35]

$$\begin{aligned} A_i \bullet X &= b_i, \quad i = 1, \dots, m, \quad X \succeq 0; \\ \sum_{i=1}^m A_i^T y_i + Z &= C, \quad Z \succeq 0; \\ XZ &= 0. \end{aligned} \tag{1.5}$$

Нелінійна система (1.5) зі змінними (X, Z, y) має $n(n+1) + m$ змінних та $m + n(n+1)/2 + n^2$ рівнянь (з рівності $X \bullet Z = 0$ слідує рівність $XZ = 0$ для додатно визначених матриць [35], причому добуток напіввизначених матриць XZ не є симетричною матрицею). Система рівнянь (1.5) без умов $X \succeq 0$, $Z \succeq 0$ має неєдиний розв'язок. Умову $X \succeq 0$ забезпечують за допомогою штрафу

$C \bullet X - \mu \ln(\det X)$, який визначений тільки для $X \succ 0$. Якщо початкова матриця $X \succ 0$, то ця умова буде зберігатися для даного штрафу.

Теорема 1.4 [109]. *Необхідні умови оптимальності для задачі зі штрафом призводять до наступної нелінійної системи рівнянь*

$$\begin{aligned} A_i \bullet X &= b_i, i = 1, \dots, m, \\ \sum_{i=1}^m A_i^T y_i + Z &= C, \\ XZ &= \mu I, \end{aligned} \quad (1.6)$$

яка при фіксованому $\mu > 0$ та початковому $X \succeq 0$ має єдиний розв'язок.

Систему (1.6) розв'язують методом Ньютона для фіксованого значення параметра $\mu > 0$, потім перераховують значення μ так, щоб $\mu \rightarrow 0$. На кожній ітерації метода Ньютона розв'язується лінійна система (дужками замінимо індекси $A_i \bullet X = A(X)$, $i = 1, \dots, m$, $A^T(y) = \sum_{i=1}^m A_i^T y_i$)

$$\begin{aligned} A(\Delta X) &= b - A(X), \\ \Delta Z + A^T(\Delta y) &= C - Z - A^T(y), \\ \Delta XZ + X\Delta Z &= \mu I - XZ, \end{aligned}$$

де шуканими є матриці ΔX , ΔZ та вектор Δy .

Виражаючи змінні, отримаємо рівняння

$$A(XA^T(\Delta y)Z^{-1}) = A(XF_2Z^{-1}) - \mu A(Z^{-1}) + b. \quad (1.7)$$

де $F_2 = C - Z - A^T(y)$. Рівняння (1.7) містить тільки один невідомий вектор Δy .

Після знаходження цього вектора знаходимо шукані матриці $\Delta Z = F_2 - A^T(\Delta y)$

та $\Delta X = \mu Z^{-1} - X\Delta ZZ^{-1} - X$.

Результатом проведених операцій може бути несиметрична матриця ΔX .

Її необхідно перетворити на симетричну наступним чином: $\Delta X = (\Delta X + \Delta X^T)/2$.

Перехід у нову точку здійснюється наступною ітераційною процедурою

$$X = X + \alpha_p \Delta X, \quad y = y + \alpha_d \Delta y, \quad Z = Z + \alpha_d \Delta Z,$$

де константи α_p та α_d належать відрізьку $(0,1]$ і обираються такими, щоб X та Z були додатно визначеними. Параметр μ на кожній ітерації перераховується за формулою $\mu = XZ/n$ [35].

Розглянута формула симетризації матриці X може порушити збіжність методу Ньютона, тому були запропоновані різні модифікації рівняння $ZX = \mu I$. У роботі [109] запропонований оператор $H_p(XZ) = \frac{1}{2}[PXZP^{-1} + P^{-T}Z^T X^T P^T] = \mu I$, який змінює третє рівняння системи Ньютона (1.6) на $H_p(\Delta XZ + X\Delta Z) = \mu I - H_p(XZ)$, або

$$P(\Delta XZ + X\Delta Z)P^{-1} + P^{-T}(Z\Delta X + \Delta ZX)P^T = \mu I - P(XZ)P^{-1} - P^{-T}(ZX)P^T.$$

Запропоновано різні види матриці P . Найбільш ефективною є матриця P , обумовлена формулою NP [95] $P = (X^{1/2}(X^{1/2}ZX^{1/2})^{-1/2}X^{1/2})^{-1/2}$, для якої оцінка числа ітерацій методу внутрішньої точки дорівнює $3mn^3 + 0.5m^2n^2$ [95].

Таким чином, алгоритм методу внутрішньої точки має наступний вигляд:

Вхідні дані: матриці C та A , розмірності $(n \times n)$, вектор-стовпчик b розмірності m , константа $\varepsilon > 0$, початкові точки – додатно визначені матриці X і Z розмірності $(n \times n)$ та вектор-стовпчик y розмірності m .

Вихідні дані: додатно визначені матриці X та Z , вектор-стовпчик y .

Крок 1. Обчислити параметр μ за формулою $\mu = XZ/n$.

Крок 2. Знайти Δy з рівняння $A(XA^T(\Delta y)Z^{-1}) = A(XF_2Z^{-1}) - \mu A(Z^{-1}) + b$, де $F_2 = C - Z - A^T(y)$.

Крок 3. Знайти $\Delta Z = F_2 - A^T(\Delta y)$ та $\Delta X = \mu Z^{-1} - X\Delta Z Z^{-1} - X$.

Крок 4. Симетризувати ΔX .

Крок 5. Перейти у нову точку за наступною ітераційною процедурою: $X = X + \alpha_p \Delta X$; $y = y + \alpha_d \Delta y$; $Z = Z + \alpha_d \Delta Z$, де константи α_p та α_d належать відрізку $(0,1]$ і обираються такими, щоб X та Z були додатно визначеними.

Крок 6. Підрахувати $\mu = XZ/n$ та, якщо $\mu > \varepsilon$, перейти на крок 2.

Отже, розглянутий метод внутрішньої точки для розв'язування задачі (1.1) є ефективним для задач напіввизначеного програмування середньої розмірності [40]. Але він застосовний тільки в тому випадку, якщо розрив двоїстості дорівнює нулю та матриці обмежень лінійно незалежні. Також допускається, що умова Слейтера (теорема 1.2) виконується.

На практиці знаходження строго допустимої початкової точки для методу внутрішньої точки є досить складною задачею. Тому були знайдені методи внутрішньої точки з недопустимою початковою точкою поліноміальної збіжності [34, 61, 67, 82, 104]. Започаткували цей напрямок у методах внутрішньої точки Nesterov Y.E. та Todd M.J. [95], але роботи по вдосконаленню цих методів продовжуються.

Використаємо схему, подану у [82]. Принцип цього методу полягає в тому, щоб розв'язати нелінійну систему (1.6) методом Ньютона, починаючи з будь-якої додатної точки ($X > 0, Z > 0, y \in R^m$), не обов'язково допустимої для задачі (1.1). Щоб досягти допустимості та оптимальності, необхідно визначити функцію оцінювання:

$$\varphi(X, y, Z) = X \bullet Z + r(X, y, Z), \quad (1.8)$$

$$\text{де } r(X, y, Z) = \|A' \bullet X - b\| + \left\| \sum_{i=1}^m y_i A_i + Z - C \right\|;$$

$$A' \bullet X = (A_1 \bullet X, A_2 \bullet X, \dots, A_m \bullet X)^T;$$

$$b = (b_1, b_2, \dots, b_m)^T.$$

У (1.8) r забезпечує допустимість, а $X \bullet Z$ (зазор двоїстості) контролює оптимальність. Мета методу полягає в тому, щоб зробити значення функції (1.8) наближеним до нуля у процесі ітерацій.

Після застосування методу Ньютонa до системи (1.6) з недопустимою початковою точкою ($X > 0$, $Z > 0$, $y \in R^m$) та при $\mu = (X \bullet Z)/n > 0$, отримаємо наступні рівняння відносно ΔX , Δy та ΔZ :

$$\begin{cases} X\Delta ZZ^{-1} + \Delta X = \sigma\mu Z^{-1} - X, & 0 < \sigma < 1, \\ A_i \bullet \Delta X = b_i - A_i \bullet X, & i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^m y_i A_i - Z. \end{cases} \quad (1.9)$$

У системі рівнянь (1.9) очевидно, що ΔZ є симетричною (відповідно до третьої рівності у системі). Проте, ΔX є не обов'язково симетричною, тому що $X\Delta ZZ^{-1}$ може бути несиметричною. Для симетризації матриці ΔX застосуємо схему симетризації Нестерова-Тода [95], яка полягає у заміні $X\Delta ZZ^{-1}$ у першій рівності на $P\Delta ZP^T$, де $P = X^{1/2}(X^{1/2}ZX^{1/2})^{-1/2}X^{1/2} = Z^{-1/2}(Z^{1/2}XZ^{1/2})^{1/2}Z^{-1/2}$.

Тоді систему (1.9) можна перетворити до наступного вигляду:

$$\begin{cases} P\Delta ZP^T + \Delta X = \sigma\mu Z^{-1} - X, & 0 < \sigma < 1, \\ A_i \bullet \Delta X = b_i - A_i \bullet X, & i = 1, \dots, m, \\ \sum_{i=1}^m \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^m y_i A_i - Z. \end{cases} \quad (1.10)$$

З системи (1.10) знаходимо ΔX , Δy та ΔZ , та робимо наступну ітерацію зі значеннями $(X, y, Z) = (X, y, Z) + \alpha(\Delta X, \Delta y, \Delta Z)$, де α обирається таким чином, щоб виконувалися умови $X > 0$, $Z > 0$ та щоб функція оцінювання (1.8) зменшилася. Виконання ітерацій закінчується, коли $\varphi < \varepsilon$.

1.3 Інші методи для розв'язування задач напіввизначеної оптимізації

Окрім методів внутрішньої точки існують й інші методи для розв'язування задач напіввизначеної оптимізації, але вони не отримали широкого розповсюдження. До непопулярних методів через їх неефективність можна віднести також наступні методи: метод спектрального добутку ([40]), метод квадратичного регулювання ([81]), метод лінійної релаксації ([80]), метод гілок та границь ([69]), методи першого та другого порядку ([70]), метод предиктора-коректора з допустимою та недопустимою початковими точками ([58, 59]), напівнескінченну релаксацію задач SDP ([80]), методи відсікаючих площин [36, 52], метод розкладання на множники ([20]). Для розв'язування задач великої розмірності останнім часом пропонуються методи розв'язування двоїстої задачі з використанням розширеного лагранжіана [40, 48, 70, 81, 93, 101, 110]. Дослідники з Росії також в останні роки почали приділяти увагу напіввизначеній оптимізації та пропонувати свої методи розв'язання задач SDP, але ці методи не набули популярності [111, 117, 118].

Визнавши практичні недоліки класичних методів внутрішньої точки, окремі дослідники запропонували альтернативні підходи для розв'язування задач SDP. У [20] розглядається використання методів першого порядку для розв'язування нелінійного переформулювання задачі SDP, отриманого за допомогою заміни додатної напіввизначеної змінної відповідним розкладанням на множники $X = VV^T$, де V – будь-яка дійсна матриця розмірності $n \times n$, не обов'язково симетрична. Цей метод не набув широкої популярності, тому що досі не знайдені ефективні методи розв'язування нелінійної неопуклої задачі. Але слід зазначити, що цей метод все ж таки намагаються пристосувати до практичних задач – він використовується у пакеті SDPLR [21].

У зв'язку з усім вищесказаним пошуки ефективніших методів тривають. Було декілька спроб узагальнити симплекс-метод для розв'язування задач SDP, але ці спроби мали тільки теоретичний характер, тому не існує аналогічних програмних пакетів для цих узагальнень [52].

1.4 Застосування SDP

1.4.1 Загальні квадратичні задачі

Велика кількість оптимізаційних задач може бути описана за допомогою загальних квадратичних функцій, тобто функцій, що визначаються довільними симетричними матрицями. Це оптимізаційні задачі з булевими змінними, задачі теорії графів, комп'ютерної геометрії та багато інших [3, 5, 31, 35, 43, 46, 54-56, 57, 74, 75, 88, 90, 96, 107, 110]. Загальній задачі квадратичної оптимізації останнім часом надається значна увага [35, 43, 55]. Використовуючи напіввизначену релаксацію, загальні квадратичні задачі зводяться до задач напіввизначеної оптимізації [43].

Загальна задача квадратичної оптимізації (QP) має наступний вигляд:

$$\min \left\{ x^T Q x + d^T x \mid x^T A_i x + b_i^T x \leq c_i, i = 1, \dots, m \right\}, \quad (1.11)$$

де Q та A_i – симетричні матриці ($n \times n$);

b_i та d – вектор-стовпчики розмірності n ;

c – вектор-стовпчик розмірності m з компонентами (c_1, \dots, c_m) ;

x – вектор-стовпчик змінних розмірності n .

Якщо матриця Q або A_i – додатно напіввизначена, то відповідна квадратична функція є опуклою. Якщо матриця Q та всі матриці A_i – додатно напіввизначені, то задача (1.11) відноситься до опуклого програмування і розв'язується за поліноміальний час. Наприклад, метод спряжених градієнтів для задачі безумовного опуклого квадратичного програмування збігається за n кроків [159].

У загальному випадку для розв'язування квадратичної задачі (1.11) можуть використовуватися методи для розв'язування задач нелінійної оптимізації, але вони не дають бажаної точності розв'язку. В [170] описані методи пошуку оцінок розв'язку задачі (1.11), в основі яких лежить метод еліпсоїдів та субградієнтні методи. Пропонуються також інші методи. Серед детермінованих методів це такі: Outer approximation, Branch and bound scheme,

Reformulation-linearization techniques, D.C. decomposition and convex envelope techniques, Duality bound techniques. Розроблено декілька видів методів випадкового пошуку [162]: метод проб з покращенням, метод імітаційного відпалу, генетичний алгоритм, метод диференціальної еволюції, метод електромагнетизму. Але всі ці методи не можуть бути використані для розв'язування прикладних задач, окрім задач малої розмірності, тому що ці методи потребують експоненційного часу розв'язання або отримують розв'язки, далекі від оптимальних.

Останнім часом наукова спільнота звернула увагу на напіввизначену релаксацію для цього класу задач [11, 43, 55]. Напіввизначена релаксація розглядається як перспективний метод не тільки для квадратичного програмування, а й для широкого класу неопуклих задач оптимізації [43].

Більш загальні поліноміальні задачі можуть бути перетворені до квадратичних (1.11) послідовною заміною $z_i = x_i^2$ та $z_i = x_i x_j$. Наприклад, поліноміальне обмеження $x_1 x_2 x_3 - x_1^5 x_2 - x_3^4 \leq 1$ перетворюється у еквівалентне квадратичне обмеження

$$\{x, z \mid z_1 z_3 - z_2 z_4 - z_3^2 \leq 1, z_1 = x_1 x_2, z_2 = x_1^2, z_3 = x_3^2, z_4 = z_1 z_2, z_2 \geq 0, z_3 \geq 0\}.$$

1.4.2 Напіввизначена релаксація

У напіввизначеній релаксації використовується перетворення квадратичного виразу $x^T A x$ до скалярного добутку матриць $A \bullet x x^T$ або $A \bullet X$ [18], де X – додатно напіввизначена матриця рангу одиниця. Таким чином, квадратичний вираз зводиться до лінійного відносно матриці X .

Розглянемо загальну задачу квадратичної оптимізації (1.11). Використаємо напіввизначену релаксацію [64] для розв'язування квадратичної задачі (1.11). Введемо матрицю змінних Y , яка може бути виражена через будь-який вектор x таким чином [43]:

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}. \quad (1.12)$$

Таким чином, Y є симетричною та додатно напіввизначеною матрицею [43]. Використовуючи матрицю Y (1.12), перепишемо цільову функцію та функції обмежень задачі (1.11):

$$x^T Q x + d^T x = \begin{pmatrix} 0 & \frac{d^T}{2} \\ \frac{d}{2} & Q \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} = \begin{pmatrix} 0 & \frac{d^T}{2} \\ \frac{d}{2} & Q \end{pmatrix} \bullet Y,$$

$$x^T A_i x + b_i^T x - c_i = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet Y.$$

Введемо позначення

$$\tilde{Q} = \begin{pmatrix} 0 & \frac{d^T}{2} \\ \frac{d}{2} & Q \end{pmatrix}, \quad \tilde{A}_i = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix}.$$

Тоді маємо таку задачу напіввизначеної оптимізації для задачі квадратичного програмування (1.11):

$$\min \left\{ \tilde{Q} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, \quad i=1, \dots, m, \quad Y \succeq 0, \quad \text{rank} Y = 1 \right\}. \quad (1.13)$$

Таке перетворення (1.12–1.13) дозволяє звести загальну квадратичну задачу (1.11) до лінійної задачі напіввизначеної оптимізації (1.13), в якій невідомою є додатно напіввизначена матриця. Відмітимо, що задача (1.13) є

опуклою незалежно від опуклості/ввігнутості задачі (1.11). Перетворена задача (1.13) буде еквівалентною до задачі (1.11), якщо матриця Y – додатно напіввизначена рангу одиниця [55]. Однак умову, що матриця повинна бути рангу одиниця, неможливо задати аналітично. Тому задачу (1.13) розв’язують без цієї умови. Таке послаблення обмежень задачі (1.13) називається *напіввизначеною релаксацією*. Тоді отримуємо таку задачу:

$$\min \{ \tilde{Q} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, i=1, \dots, m, Y \succeq 0 \}. \quad (1.14)$$

У загальному випадку розв’язок задачі (1.14) визначить нижню оцінку цільової функції задачі (1.11) (видалення обмеження на ранг матриці з задачі (1.13) розширює або залишає незмінною допустиму множину).

Таким чином, на відміну від інших методів розв’язування загальних квадратичних задач, при використанні напіввизначеної релаксації можливо точно сказати, чи є знайдений розв’язок задачі (1.14) глобальним мінімумом задачі (1.11). Якщо ранг матриці Y дорівнює одиниці, то знайдений розв’язок є глобальним мінімумом задачі (1.11). Якщо ранг матриці Y більше одиниці, то буде отримано нижню оцінку цільової функції задачі (1.11).

1.4.3 Максимум норми вектора на перетині еліпсоїдів

Напіввизначена релаксація дозволяє апроксимувати неопуклі допустимі множини опуклими. Якщо перетворити неопуклі множини на опуклі, то для таких множин напіввизначена релаксація буде давати кращі оцінки розв’язку задачі. Наприклад, для опуклого многогранника така апроксимація буде точною.

В роботі [124] обґрунтовано новий метод для розв’язування нелінійних багатоекстремальних задач. Показано, що розв’язування таких задач зводиться до знаходження мінімуму або максимуму квадрата норми векторf на опуклій множині. Якщо задача квадратична, то перетворена задача може бути розв’язана методом напіввизначеної оптимізації. Розглянемо суть методу точної квадратичної регуляризації для розв’язування багатоекстремальних задач виду

$$\min\{ f_0(x) \mid f_i(x) \leq 0, i = 1, \dots, m, x \in R^n \}, \quad (1.15)$$

де $f_i(x)$, $i = 0, 1, \dots, m$ – двічі диференційовані функції;

R^n – евклідовий простір.

Припустимо, що розв'язок задачі (1.15) існує. Введемо нову змінну x_{n+1} та зведемо задачу (1.15) до вигляду

$$\min\{ x_{n+1} \mid f_0(x) + s \leq x_{n+1}, f_i(x) \leq 0, i = 1, \dots, m \}, \quad (1.16)$$

де значення параметру s вибираємо таким, щоб

$$f_0(x^*) + s \geq \|x^*\|^2, \quad (1.17)$$

x^* – розв'язок задачі (1.15). Для визначення параметра s достатньо вибрати мінімальне значення $\|x^*\|^2$ на множині точок глобального мінімуму. Значення параметра s , що задовольняє умові (1.17) завжди існує, так як цільова функція задачі (1.15) обмежена знизу на допустимій множині.

Після розв'язуванні задачі (1.16) перевіряємо умову (1.17), якщо вона не виконується, то збільшуємо значення параметра s . Таке збільшення продовжується до виконання умови (1.17). Далі, перетворенням простору $x = Az$, де матриця A порядку $(n+1) \times (n+1)$ має вигляд

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ z_1 & z_2 & \dots & z_{n+1} \end{pmatrix},$$

задачу (1.16) зведемо до наступної задачі

$$\min\{ \|z\|^2 \mid f_0(\bar{z}) + s \leq \|z\|^2, f_i(\bar{z}) \leq 0, i = 1, \dots, m, z \in E^{n+1} \}, \quad (1.18)$$

де $\bar{z} = (z_1, \dots, z_n)$, $z = (\bar{z}, z_{n+1})$.

Існує таке значення параметра $r > 0$, що всі функції

$$g_0(z) = f_0(\bar{z}) + s + (r-1)\|z\|^2, g_i(z) = f_i(\bar{z}) + r\|z\|^2, i = 1, \dots, m,$$

будуть опуклими. Дійсно, при відповідному виборі параметра $r > 0$, гессіани функцій $g_i(z)$, $i = 0, \dots, m$ – це матриці з переважаючою головною діагоналлю, тобто додатно визначені (якщо серед $f_i(\bar{z}), i = 1, \dots, m$ є опуклі функції, то ці обмеження залишаються незмінними).

Таким чином, задача (1.18) зведена до наступної

$$\min\{\|z\|^2 \mid g_i(z) \leq d, i = 0, \dots, m, r\|z\|^2 = d\}, \quad (1.19)$$

де всі $g_i(z)$ – опуклі функції;

d – нова змінна.

Отже, задача (1.15) перетворена до мінімізації квадрата норми вектора z , де змінними задачі (1.19) є (z, d) – $(n+2)$ -вектор.

Теорема 1.5.[124] *Якщо (z^*, d^*) – розв'язок задачі (1.19) і для параметра s виконується умова (1.17), тоді $x^* = \bar{z}^*$ – розв'язок задачі глобальної оптимізації (1.15).*

Розглянемо дві множини $S_1(d) = \{z \mid g_i(z) \leq d, i = 1, \dots, m\}$ та $S_2(d) = \{z \mid r\|z\|^2 \leq d\}$. Якщо $S_1(d_m) \cap \text{int } S_2(d_m) \neq \emptyset$, де d_m – розв'язок опуклої задачі

$$\min\{d \mid g_i(z) \leq d, i = 0, \dots, m, r\|z\|^2 \leq d\},$$

то задача (1.19) є еквівалентною задачі максимізації квадрата норми вектора на опуклій множині

$$\max\{\|z\|^2 \mid g_i(z) \leq d, i = 0, \dots, m\}. \quad (1.20)$$

В цій задачі необхідно знайти мінімальне значення d , при якому множина $S_1(d)$ торкається границі множини $S_2(d)$ зсередини. При менших значеннях d допустима множина задачі (1.19) буде пустою.

Перетворена задача (1.20) має опуклу допустиму множину та краще апроксимується напіввизначеними обмеженнями, що буде використано далі.

1.4.4 Задачі булевої оптимізації

Напіввизначене програмування є дуже корисним при розв'язуванні комбінаторних задач оптимізації NP-складності [3, 5, 11, 31, 32, 35, 37, 43, 54, 56, 57, 63, 88, 96, 97, 99, 107, 108, 110]. Багато задач з різних областей знань вимагають вибору однієї з безлічі можливих альтернатив. Такі задачі виникають в економіці, управлінні, техніці, розподілі ресурсів, в проектуванні, інформаційних системах і т.п. При математичному моделюванні ці задачі перетворюються до лінійних або квадратичних задач оптимізації з булевими змінними. Більшість таких задач відносяться до класу NP-складних, для яких розроблені тільки експоненційні методи. Ці методи ефективні для розв'язування задач малої розмірності, але оптимізаційні задачі часто містять тисячі і навіть сотні тисяч змінних. Для розв'язування таких задач необхідно розробити нові, ефективніші методи. У багатьох випадках оптимальний розв'язок SDP-релаксації може бути перетворено до допустимого розв'язку початкової задачі.

Найпопулярніші задачі комбінаторної оптимізації, в яких застосовується SDP, це задача max-cut [3, 5, 31, 35, 43, 54, 56, 57, 74, 75, 88, 90, 96, 107, 110], задача розфарбовування графа мінімальною кількістю кольорів [43, 46, 56, 57] та булеве програмування [3, 43, 56, 110].

Задачі булевого програмування можуть мати лінійні або нелінійні обмеження. Напіввизначеній релаксації для таких типів задач присвячена [43].

Розглянемо задачу булевої оптимізації

$$\min\{c^T x \mid Ax \leq b, x \in R^n, x_i = 0 \vee 1\}, \quad (1.21)$$

де c – n -вимірний вектор-стовпчик;

b – m -вимірний вектор-стовпчик;

A – матриця розмірності $(m \times n)$;

x – n -вимірний вектор-стовпчик з компонентами x_i , що дорівнюють 0 або 1.

Пропонована методика розв'язування задач застосовна і для більш загальної задачі квадратичної оптимізації з квадратичними обмеженнями та булевими змінними

$$\min\{x^T Q_0 x + q_0^T x \mid x^T Q_i x + q_i^T x \leq r_i, i = 1, \dots, m, x_j = 0 \vee 1, x \in R^n\}, \quad (1.22)$$

де усі Q_i – симетричні матриці розмірності $(n \times n)$;

q_i, x – n -вимірні вектор-стовпчики;

r_i – скаляр.

Замінімо в задачах (1.21–1.22) булеві змінні квадратичною умовою $x_j(x_j - 1) = 0, j = 1, \dots, n$, де тепер змінні x_j можуть приймати довільні значення. Звичайно, що допустимими будуть тільки значення 0 або 1.

Таким чином, задачі (1.21–1.22) перетворюються на загальну квадратичну задачу

$$\min\{x^T Q_0 x + q_0^T x \mid x^T Q_i x + q_i^T x \leq r_i, i = 1, \dots, m, x_j(x_j - 1) = 0, j = 1, \dots, n, x \in R^n\}.$$

1.5 Огляд програмного забезпечення для розв'язування задач SDP

SDPA 7.0. Цей програмний пакет використовується для розв'язування загальних задач напіввизначеного програмування. Проект SDPA був запущений у 1995 році. За рахунок використання пакету для швидкої обчислювальної роботи з матрицями достатньо ефективно розв'язуються задачі великої

розмірності. Для розв'язування задач SDP в пакеті використовується прямо-двоїтий метод внутрішньої точки. SDPA 6.0 успішно розв'язує двоїсті задачі SDP з числом обмежень $m = 1500$, які інші програмні пакети не можуть розв'язати у зв'язку з нестачею пам'яті. Автори – *Katsuki Fujisawa, Masakazu Kojima, Kazuhide Nakata* [105, 106].

SDPT 3. Це програмне забезпечення було розроблено, аби розв'язувати задачі опуклої оптимізації, де в якості обмежень можуть використовуватися різні конуси: конус додатно визначених матриць, невід'ємні ортанти та Евклідовий простір; та в якості цільової функції виступає сума лінійних функцій. Для розв'язання використовується прямо-двоїтий метод внутрішньої точки. Є можливість роботи із задачами, заданими у форматі SDPA та SeDuMi. Автори – *Kim Chuan Toh, Michael J. Todd, Reha Tutuncu* [97].

CSDP. Цей програмний пакет написаний на мові C з використанням LAPACK і LINPACK – пакетів для швидкої обчислювальної роботи з матрицями. CSDP однаково ефективно обробляє як обмеження у вигляді лінійних нерівностей, так і у вигляді лінійних рівностей. Для розв'язування задач SDP використовується метод внутрішньої точки. Автор – *B. Borchers* [16].

SeDuMi. Цей програмний пакет написаний під MatLab. Використовується для розв'язування оптимізаційних задач над симетричним конусом, також дозволяє задавати не тільки лінійні, але і квазіопуклі квадратичні та додатно визначені обмеження. В даному програмному продукті збалансована швидкість і точність обчислень. Автор – *Jos F. Sturm* [94].

DSDP. У цьому програмному пакеті реалізований прямо-двоїтий метод для розв'язування задач SDP. DSDP може використовуватися як окрема бібліотека підпрограм, як функція в середовищі MatLab, або як виконуваний файл, який одержує дані з файлу та записує результат обчислень теж у файл. Перша версія DSDP була випущена для розв'язування задач комбінаторної оптимізації методом напіввизначеної релаксації. Надалі програма удосконалювалася і в даний момент з її допомогою можна розв'язувати загальні задачі напіввизначеного програмування. Автори – *Steve Benson, Yinyu Ye, Xiong Zhang* [10].

CUTSDP. Це пакет підпрограм, які містять реалізацію методу відсікаючих площин, що базується на SDP. На даний момент може використовуватися для розв'язування задачі пошуку максимального розрізу графа, а також для задач поділу графів на два рівні підграфи. Автор – *Stefan E. Karisch*.

SDPPACK 0.9. Працює на платформі MatLab 5.0. За допомогою цієї версії можна розв'язувати пряму та двоїсту задачі SDP, задачі напіввизначеного або лінійного квадратичного програмування, у тому числі задачі лінійної оптимізації над додатно визначеним конусом, квадратичним конусом та невід'ємним ортантом. Головна підпрограма працює на прямо-двоїстому методі предиктора-коректора. Експерименти показують, що код забезпечує високу точність розв'язків широкого кола задач [84]. Автори – *Farid Alizadeh, Michael L. Overton, Stefan Schmieta* [84].

SDPLIB. Це збірка підпрограм для розв'язування задач напіввизначеного програмування та задач, які зводяться до SDP. Поточна версія бібліотеки дозволяє розв'язувати 92 види задач SDP, які впливають з багатьох різних прикладних задач напіввизначеного програмування. Вхідними даними є задачі у sparse-форматі. Автор – *Brian Borchers* [17].

SDPSOL. Є одночасно парсером (програма, що здійснює граматичний розбір) та програмою для розв'язування задач SDP та задач MAXDET. Вхідні дані вводяться на мові *sdpsol*. Задачі SDP розв'язуються методом внутрішньої точки, результат видається в зручній користувачеві формі. Головною ідеєю мови *sdpsol* є дати користувачу можливість вводити дані, максимально наближені до природного математичного вигляду постановки задачі, наприклад, задавати матричні змінні у вигляді матриць, а не у вигляді векторів, як це реалізується у всіх інших описаних вище пакетах. Автори – *Shao Wu, Stephen P. Boyd* [103].

YALMIP. Використовується для розв'язування складних задач оптимізації. Розповсюджується як вільна бібліотека до MatLab або як окремий пакет. YALMIP не є однонаправленою програмою, цей пакет може використовуватися для розв'язування великого спектру оптимізаційних задач, наприклад, лінійне, квадратичне, напіввизначене, булеве, мультипараметричне програмування.

Автор – *Johan Loefberg*.

SP. Перша версія цієї програми була випущена в 1994 році та була першим вільно розповсюджуваним пакетом для розв'язування задач SDP. Реалізує прямо-двоїстий метод внутрішньої точки Нестерова і Тодда для задач SDP [96]. Включає повне C-джерело, яке може бути використано безпосередньо або через інтерфейс MatLab MEX, MatLab приклади та документацію. Автори – *Lieven Vandenberghe, Stephen P. Boyd, Brien Alkire* [91].

PENNON. Ця комп'ютерна програма призначена для розв'язування задач опуклої і неопуклої нелінійної оптимізації та напіввизначеного програмування. Вона є платною. Цей програмний пакет спрямований на задачі великої розмірності. Дані для розв'язання повинні бути надані у sparse-форматі. Ця програма базується на узагальненому методі розширеного лагранжіану. Автор – *Michal Kocvara* [48, 49, 93].

PENSDP. Платний пакет. Розробники заявляють, що цей пакет є одним з найбільш ефективних солверів для задач великої розмірності. Використовується для розв'язування лінійних задач напіввизначеного програмування. Може використовуватися як окрема програма, як бібліотека пакету MatLab або як функція у C++ або Fortran. Автори – *Michal Kocvara, Michael Stingl* [50].

SDPLR. Цей пакет написаний на мові C. Використовується для розв'язування задач напіввизначеної оптимізації великої розмірності. Підтримується інтерфейс MatLab. Автори – *Samuel Burer, Renato D.C. Monteiro, Changhui Choi* [21].

NEOS SERVER (<http://neos.mcs.anl.gov/neos/solvers/index.html>). Це інтернет-сервер для розв'язування різних видів задач оптимізації: комбінаторної, цілочисельної, глобальної, лінійної, напіввизначеної, нелінійної, напівнескінченної. У випадку напіввизначеної оптимізації на цьому сервері є можливість розв'язати задачі SDP за допомогою солверів CSDP, DSDP, PENSDP, SDPA, SDPLR, SDPT3, SEDUMI. NEOS є відкритим каталогом, тобто розробники власних солверів можуть самостійно їх підключати.

1.6 Симплекс-метод для задач лінійної оптимізації

Напіввизначений симплекс-метод для розв'язування задач напіввизначеної оптимізації, який обґрунтовується в даній роботі, базується на симплекс-методі для розв'язування задач лінійного програмування. Симплекс-метод для розв'язування задач лінійного програмування був запропонований Дж. Данцигом в 1947 році [113]. Фактично цей метод реалізується тільки на ЕОМ, за винятком задач малої розмірності. Суть методу полягає в тому, як перейти від поточного базису до базису з меншим значенням цільової функції і в перевірці поточного базису на оптимальність. Перехід від одного базису до іншого здійснюється шляхом заміни однієї з базисних змінних іншою – небазисною.

Наведемо основні положення симплекс-методу для задач лінійного програмування, які будуть використані для побудови напіввизначеного симплекс-методу.

Будь-яка задача лінійного програмування легко перетворюється до канонічного вигляду

$$\min\{c^T x \mid Ax = b, x \geq 0, x \in R^n\},$$

де c – вектор-стовпчик розмірності n ;

b – вектор-стовпчик розмірності m ;

A – матриця розмірності $(m \times n)$;

x – вектор-стовпчик змінних розмірності n .

Задача лінійного програмування містить n змінних і m обмежень (не враховуючи умов додатності змінних). Як правило, у задачі лінійного програмування лінійна система має незліченну кількість розв'язків, з яких необхідно вибрати один, який мінімізує значення цільової функції. Якщо $n - m$ змінних прирівняти до нуля, то усічена система матиме єдиний розв'язок за певних умов. Такими умовами є невиродженість одержуваної системи рівнянь і, крім того, її розв'язок має бути невід'ємним. Отриманий розв'язок, якщо всі

його змінні невід'ємні, називається *базисним розв'язком*, або *базисом*.

Деякі зі змінних базису можуть бути нульовими, тоді такий базис називається виродженим. Число базисних розв'язків скінченне і не перевищує числа комбінацій з n по m .

Теорема 1.6. [124] *Множина крайніх точок многогранника $\{x \mid Ax = b, x \geq 0\}$ відповідає множині базисних розв'язків.*

Теорема 1.7. [124] *Оптимальний розв'язок задачі лінійного програмування досягається, принаймні, в одній крайній точці.*

З наведених теорем випливає, що для знаходження оптимального розв'язку задачі лінійного програмування необхідно знайти початковий базисний розв'язок і потім вказати алгоритм переходу до іншого базисного розв'язку з меншим значенням цільової функції. Для завершення пошуку потрібно ідентифікувати базисний розв'язок, який є оптимальним розв'язком задачі.

У загальному випадку для пошуку початкового базису використовується метод штучного базису [124]. Оптимальним розв'язком такої задачі буде рівність нулю всіх штучних змінних. Якщо це не так, то допустима множина буде порожньою. Складнощі можуть виникнути, коли розв'язок задачі мінімізації суми штучних змінних буде виродженим. Тоді нульові штучні змінні будуть залишатися в базисі і необхідно мінімізувати початкову цільову функцію при знайденому базисі. Доведено, що нульові штучні змінні не можуть стати додатними [113].

Опишемо кроки симплекс-метода.

Крок 1. Виключаємо в рядку цільової функції коефіцієнти при базисних змінних, використовуючи процедуру виключення Гауса. Потім перевіряємо, чи є в перетвореному рядку цільової функції значення, менші від нуля. Якщо таких значень немає, то поточний базис буде оптимальним (розв'язок дорівнює $x_B = b$). В іншому випадку переходимо до наступного кроку.

Крок 2. Знаходимо у перетвореному рядку цільової функції мінімальний елемент. Відповідну змінну вводимо в базис, а відповідний стовпчик таблиці будемо називати провідним. Визначимо, яку змінну необхідно виключити з базису. Змінна, яка виключається з базису, визначається з умови, що

перетворення нового базису (матриці B до одиничної) повинно залишати праві частини обмежень невід'ємними. Ця змінна визначається з умови

$$\min_{a_{is} > 0} \left\{ \frac{b_i}{a_{is}} \right\} = \frac{b_r}{a_{rs}}. \text{ Мінімум може досягатися при декількох значеннях } i, \text{ тоді}$$

наступний базис буде виродженим. Відповідний рядок називають провідним, а елемент таблиці, що стоїть на перетині провідного рядка і стовпчика – ведучим. Перетворимо симплекс-таблицю таким чином, щоб ведучий елемент дорівнював одиниці, а всі інші елементи провідного стовпця – нулю. В результаті отримаємо новий базисний розв'язок. Перейдемо до кроку 1.

Зауваження. При гаусових перетвореннях симплекс-таблиці останній стовпець повинен весь час бути додатнім.

При розв'язуванні задачі лінійного програмування симплекс-методом можливо 4 результати: буде отриманий єдиний розв'язок, буде отриманий розв'язок, але не єдиний, допустима множина і цільова функція необмежені, допустима множина порожня. Симплекс-метод дозволяє визначити кожен з цих результатів. Якщо оптимальний базис отриманий і коефіцієнти в цільовій функції при небазисних змінних (при базисних змінних вони дорівнюють нулю) є додатніми, то отриманий розв'язок єдиний. Якщо ж є нульові коефіцієнти і відповідний їм стовпець перетвореної матриці обмежень одиничний, то задача має незліченну кількість розв'язків з одним і тим же значенням цільової функції (розв'язок знаходиться на одній з граней многогранника). Якщо на якійсь ітерації симплекс-методу всі компоненти ведучого стовпця недодатні, то цільова функція необмежена. Нарешті, якщо з поточного базису не вдається виключити всі штучні змінні і вони не дорівнюють нулю, то допустима множина задачі є порожньою і задача лінійного програмування не має жодного розв'язку.

Критерієм зупинки (знаходження оптимального розв'язку задачі лінійного програмування) в симплекс-методі є невід'ємність перетворених коефіцієнтів цільової функції. Цей критерій також буде використано в напіввизначеному симплекс-методі.

Симплекс-метод довів свою практичну ефективність (для досягнення

точки мінімуму потрібно в середньому близько $3m$ ітерацій), однак не виключені можливості зациклення [124].

Теорема 1.8 (Теорема Бленда) [124]. Нехай ведучий стовпець для вводу в базис у симплекс-алгоритмі вибирається за правилом $j = \min\{j \mid c'_j < 0\}$, а ведучий рядок з умови $i = \min\{i \mid \min(b'_i / a'_{ki} \mid a'_{ki} > 0)\}$, тоді симплекс-алгоритм закінчує роботу після кінцевого числа ітерацій.

Для розв'язування задач великої розмірності для переходу від поточного базису до базису з меншим значенням цільової функції може використовуватися метод генерації стовпців [144]. Такий прийом називається генерацією стовпців, так як при розв'язуванні підзадачі фактично використовується лише невелика кількість стовпців множини, які генеруються по мірі необхідності.

1.7 Методи знаходження власних значень матриці та відповідних їм власних векторів

Для побудови напіввизначеного симплекс-методу крім розглянутого симплекс-методу для лінійного програмування потрібна також ефективна процедура знаходження мінімуму квадратичної функції $x^T Q x$ за умови $\|x\|^2 = 1$, де Q – симетрична матриця (ця процедура визначає додатну напіввизначеність матриці Q). Ця задача рівносильна знаходженню власного вектора матриці Q , що відповідає її мінімальному власному значенню. Існує багато методів для знаходження власних векторів матриць, що зумовлено складністю чисельної реалізації цієї нелінійної задачі. Розглянемо ці методи.

Задачам на власні числа присвячено величезну кількість публікацій [39, 79, 114, 120, 150, 151, 165]. Це пов'язано з тим, що багато задач механіки та інших прикладних задач призводять до необхідності знаходження власних чисел. У квантовій хімії та молекулярній динаміці взаємодія електронів в молекулах та їх рух визначається рівнянням Шредінгера, в якому виникає задача пошуку власних чисел і власних функцій. У фізиці одним із завдань хвильової оптики є пошук розв'язку так званого векторного хвильового рівняння для електричного

поля, для розв'язання якого потрібно знайти власні числа [79]. Багато задач на власні значення виникає в технічній механіці: вирішення проблем стійкості механічних систем, їх коливань і багато інших [114].

Проблема обчислення власних значень є складною для погано обумовлених матриць і матриць великого розміру. Існує багато методів для знаходження власних значень, кожен з яких має свої переваги [39, 114, 151, 165]. Детальний аналіз чисельної збіжності методів наведений у книгах [114, 151, 165] та у [131].

На даний час найкращим методом для знаходження всіх власних значень матриці A є QR-алгоритм [165, 39], проте його програмна реалізація є досить складною. У методі Якобі [114] по заданій симетричній матриці $A = A_0$ будується послідовність ортогонально подібних матриць A_1, A_2, \dots , яка зрештою збігається до діагональної матриці, на діагоналі якої знаходяться власні значення. Матриця A_{i+1} отримується з A_i по формулі $A_{i+1} = J_i^T A_i J_i$, де J_i – ортогональна матриця, яка називається обертянням Якобі. При відповідному виборі обертань стовпці матриці J будуть наближеними власними векторами матриці A [114]. Зазвичай метод Якобі працює набагато повільніше, ніж конкуруючі методи, і продовжує становити інтерес тільки з тієї причини, що іноді здатний обчислювати малі власні значення і відповідні їм власні вектори з набагато більшою точністю, ніж інші методи [114].

Крім того, в багатьох практичних задачах потрібно знайти тільки мінімальне або максимальне власне значення матриці (ця задача називається частковою проблемою власних значень). Для пошуку екстремальних власних значень матриці найчастіше використовують метод прямої або оберненої ітерації [39]. Будемо далі розглядати тільки симетричні матриці. Спочатку знаходять власний вектор, використовуючи ітераційну процедуру $x^{k+1} = Ax^k$, $k = 0, 1, \dots$. На кожному кроці вектор x^{k+1} нормується або ділиться на максимальну компоненту x^{k+1} . Вектор x^k прямує до власного вектора x , коли $k \rightarrow \infty$, при відповідному виборі початкового вектора x^0 . Тоді максимальне власне число визначається за формулою $\lambda = x^T Ax / \|x\|^2$, де $\|\cdot\|$ – означає

евклідову норму.

Для знаходження мінімального власного значення матриці A розглядається ітераційна процедура для оберненої матриці $x^{k+1} = A^{-1}x^k$, $k = 0, 1, \dots$. Метод простої ітерації збігається глобально до власного вектора з будь-якої початкової точки, крім точок гіперплощини, для якої шуканий власний вектор є нормаллю. Однак збіжність методу простої ітерації повільна. Пропонувалися різні удосконалення даного методу, але вони лише в деяких випадках дозволяють прискорити його збіжність [165]. Зокрема, використовується зсув Релея $x^{k+1} = (A - \sigma_k I)^{-1}x^k$, де I – одинична матриця, а

$$\sigma_k - \text{відношення Релея } (\sigma_k = \frac{x_k^T A x_k}{x_k^T x_k}).$$

У даній роботі удосконалення методу прямої та оберненої ітерації використовує спряжені напрями. Це пов'язано з тим, що для задач квадратичної оптимізації, якою є знаходження власного вектору, метод спряжених напрямів збігається за n ітерацій.

РОЗДІЛ 2

НАПІВВИЗНАЧЕНИЙ СИМПЛЕКС-МЕТОД ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ НАПІВВИЗНАЧЕНОЇ ОПТИМІЗАЦІЇ

У цьому розділі для визначення додатної напіввизначеності матриць пропонується модифікована процедура оберненої ітерації з використанням методу спряжених напрямів. Удосконалюється напіввизначений симплекс-метод, який використовується для розв'язування задач напіввизначеної оптимізації, шляхом використання методу спряжених напрямів у процедурі оберненої ітерації, та строго обґрунтовується його збіжність за допомогою теорем і лем. Встановлюються теоретичні переваги напіввизначеного симплекс-методу над методами внутрішньої точки; наводяться порівняльні чисельні експерименти напіввизначеного симплекс-методу з методами внутрішньої точки за допомогою власного програмного забезпечення та існуючих програмних пакетів для розв'язування задач SDP.

2.1 Метод спряжених напрямів для визначення додатної напіввизначеності симетричної матриці

Для побудови напіввизначеного симплекс-методу необхідна ефективна процедура знаходження додатної напіввизначеності матриць, яка зводиться до мінімуму квадратичної функції $x^T Q x$ за умови $\|x\|^2=1$, де Q – симетрична матриця. Ця задача рівносильна знаходженню власного вектора матриці Q , що відповідає її мінімальному власному значенню. В напіввизначеному симплекс-методі така задача буде розв'язуватися на кожній ітерації, тому потрібен надійний та швидкий метод знаходження мінімального власного значення та відповідного власного вектора матриці Q .

У цій задачі необхідно знайти одне мінімальне власне значення, тому немає необхідності використовувати методи, які знаходять усі власні значення

(метод Якобі, метод QR). Метод оберненої ітерації шукає мінімальне власне значення і має лінійну збіжність. Саме тому за основу був узятий метод оберненої ітерації. Але збіжність методу оберненої ітерації повільна. Пропонувалися різні удосконалення даного методу, але вони лише в деяких випадках дозволяють прискорити його збіжність. Далі використовується метод спряжених напрямів в методі оберненої ітерації для розв'язування цієї задачі (спочатку вона перетворюється до опуклої). Це пов'язано з тим, що для мінімізації опуклих квадратичних функцій метод спряжених напрямів є простим та ефективним, тому що збігається за n ітерацій.

2.1.1 Постановка задачі і метод розв'язування

Необхідно знайти такий власний вектор x , який задовольняє рівнянню $Qx = \lambda x$, де Q – симетрична матриця порядку n , а λ – мінімальне власне число. Для довільної симетричної матриці Q існує n власних значень, частина з яких можуть бути кратними (тобто їх значення співпадають). Власний вектор, відповідний мініимальному власному значенню, знаходимо з розв'язку задачі

$$\min\{x^T Qx \mid \|x\|^2 = 1\}. \quad (2.1)$$

Розв'язок цієї задачі еквівалентний розв'язку наступної (до цільової функції додається доданок, який дорівнює нулю)

$$\min\{x^T Qx + r(\|x\|^2 - 1) \mid \|x\|^2 = 1\},$$

або

$$\min\{x^T (Q + rI)x - r \mid \|x\|^2 = 1\}, \quad (2.2)$$

де I – одинична матриця.

Так як константа $r > 0$ (її значення визначається нижче) не впливає на значення точки мінімуму x (зміниться тільки значення цільової функції), то доданок $-r$ можна опустити. Тобто, розв'язок задачі (2.2) еквівалентний розв'язку наступної задачі

$$\min\{x^T(Q+rI)x \mid \|x\|^2=1\}.$$

Оберемо таке значення $r > 0$, при якому матриця $Q^* = Q + rI$ буде додатно визначеною. Для цього достатньо, щоб діагональні елементи матриці Q^* були більше суми модулів відповідних недіагональних елементів

$$q_{ii}^* > \sum_{i \neq j} |q_{ij}^*|, \forall i, j,$$

де q_{ij}^* – елементи матриці Q^* . Наприклад, для матриці

$$Q = \begin{pmatrix} -1 & 2 \\ 2 & 1 \end{pmatrix}$$

константу r достатньо узяти рівною 3. Отримаємо матрицю

$$Q^* = \begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix},$$

яка є додатно визначеною.

Таким чином, розв'язування задачі (2.2) зводиться до пошуку власного вектора додатно визначеної матриці Q^* , відповідного її мінімальному власному значенню. Це рівносильно розв'язуванню задачі

$$\min\{x^T Q^* x \mid \|x\|^2=1\}, \quad (2.3)$$

або, враховуючи те, що власний вектор досить знайти з точністю до постійного множника, задачі

$$\max\{\|x\|^2 \mid x^T Q^* x = 1\}. \quad (2.4)$$

Теорема 2.1. *Розв'язок задачі (2.3) співпадає з розв'язком задачі (2.4) з точністю до постійного множника.*

Доведення. Нехай x_2 – розв'язок задачі (2.4). Покажемо, що $x_2 = \alpha x_1$, де $\alpha > 0$ – деяка константа. Точка x_2 є точкою останнього дотику кулі $\{x \mid \|x\|^2 = R^2\}$ до границі еліпсоїда $\{x \mid x^T Q^* x = 1\}$ при збільшенні радіусу кулі R (рис. 2.1). Якщо $R = 1$, то ця точка буде також розв'язком x_1 задачі (2.3). Інакше змінимо пропорційно діагоналі еліпсоїда таким чином, щоб цей еліпсоїд був вписаний в кулю радіусу одиниця. Тоді точка дотику буде лежати на тій же діагоналі еліпсоїда. Це означає, що розв'язки задач (2.3) та (2.4) відрізняються на деякий множник та є власним вектором матриці Q^* . Теорему доведено.

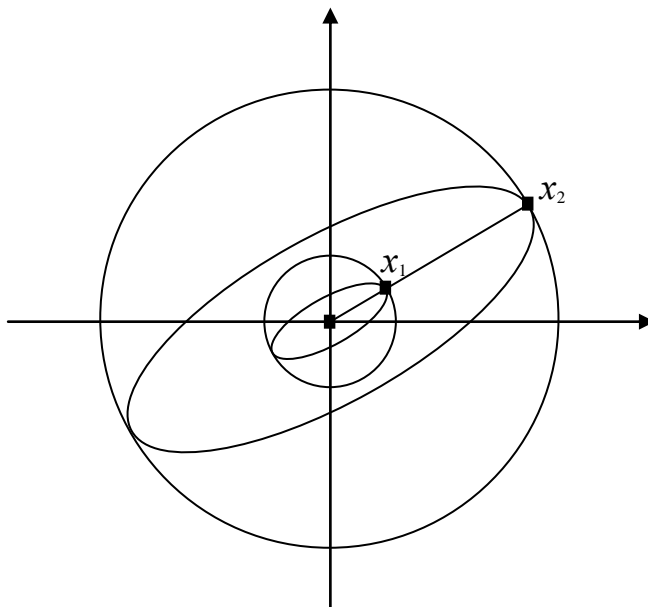


Рисунок 2.1 – Оптимальні розв'язки задач (2.3) та (2.4), де x_1 – оптимальний розв'язок задачі (2.3) та оптимальний розв'язок задачі (2.4) при $R = 1$;

x_2 – оптимальний розв’язок задачі (2.4) при $R > 1$.

Тобто для додатно визначеної матриці Q^* задача (2.4) має просту геометричну інтерпретацію. У задачі (2.4) необхідно знайти точку останнього дотику кулі $\{x \mid \|x\|^2 = R^2\}$ до границі еліпсоїда $\{x \mid x^T Q^* x = 1\}$ при збільшенні радіусу кулі R .

Таким чином, для перевірки додатної напіввизначеності матриці Q достатньо знайти власний вектор x додатно визначеної матриці Q^* , відповідний мінімальному власному значенню Q^* , та перевірити умову $x^T Q x \geq 0$. Якщо вона виконується, то матриця Q – додатно напіввизначена.

Теорема 2.2. Нехай x^* – розв’язок задачі (2.4) та $x^{*T} Q x^* \geq 0$, тоді матриця Q – додатно напіввизначена.

Доведення. Точка максимуму x^* задачі (2.4) є точкою мінімуму задачі (2.3) та задачі (2.1). Тоді з нерівності $x^{*T} Q x^* \geq 0$ випливає нерівність $x^T Q x \geq 0$ для будь-яких x . Тобто, нерівність $x^T Q x \geq 0$ виконується для всіх значень вектора x , а це є визначенням додатної напіввизначеності матриці Q . Теорема доведена.

Для розв’язування задачі (2.4) використовуємо метод множників Лагранжа, замінюючи на кожній ітерації її цільову функцію лінійною апроксимацією:

$$\max\{(x^k)^T x \mid x^T Q^* x = 1\}, \quad k = 0, 1, \dots$$

Функція Лагранжа буде мати вигляд:

$$L(x, \lambda) = (x^k)^T x + \lambda(x^T Q^* x - 1).$$

Знайдемо похідні та прирівняємо їх нулю:

$$\begin{cases} \frac{\partial L}{\partial x} = x^k + 2\lambda Q^* x = 0, \\ \frac{\partial L}{\partial \lambda} = x^T Q^* x - 1 = 0. \end{cases}$$

Тоді з першого рівняння маємо:

$$2\lambda Q^* x = -x^k,$$

$$x = -\frac{(Q^*)^{-1} x^k}{2\lambda}.$$

Підставимо цей вираз у друге рівняння:

$$\frac{(x^k)^T (Q^*)^{-1} Q^* (Q^*)^{-1} x^k}{4\lambda^2} = 1,$$

або

$$\frac{(x^k)^T (Q^*)^{-1} x^k}{4\lambda^2} = 1,$$

$$\frac{1}{2\lambda} = \pm \frac{1}{\sqrt{(x^k)^T (Q^*)^{-1} x^k}}.$$

Тоді, так як ми знаходимо $\max \|x\|^2$, знак λ можна не враховувати і достатньо взяти $\lambda > 0$, маємо

$$x^{k+1} = \frac{(Q^*)^{-1} x^k}{\sqrt{(x^k)^T (Q^*)^{-1} x^k}}, \quad (2.5)$$

звідки отримуємо рекурентну формулу для знаходження k -го наближення розв'язку задачі (2.4) у вигляді

$$x^k = \frac{(Q^*)^{-k} x^0}{\sqrt{x^{0T} (Q^*)^{-(2k-1)} x^0}},$$

де x^0 – початкова точка.

Очевидно, що отримана процедура збігається з методом оберненої ітерації для додатно визначеної матриці Q^* . Значення x^k відрізняється від розв'язку, який отримується степеневим методом, нормуванням вектора x^k на кожній ітерації.

Враховуючи, що задача (2.4) – квадратична, для прискорення збіжності методу оберненої ітерації використовуємо процедуру побудови спряжених напрямів. Два вектора x та z – спряжені, якщо $x^T Q^* z = 0$. Тепер у методі зворотної ітерації замість вектора x^{k+1} будемо брати його корекцію так, щоб новий вектор дорівнював $z^{k+1} = x^{k+1} - \alpha x^k$, де параметр α обираємо з умови, що вектори x^k та z^{k+1} – спряжені. Це означає, що

$$(x^k)^T Q^* z^{k+1} = (x^k)^T Q^* (x^{k+1} - \alpha x^k) = (x^k)^T Q^* x^{k+1} - \alpha (x^k)^T Q^* x^k = 0,$$

звідки

$$\alpha = \frac{(x^k)^T Q^* x^{k+1}}{(x^k)^T Q^* x^k}.$$

Вважаємо $x^{k+1} = z^{k+1}$ і на наступній ітерації використовуємо формулу (2.5). Шуканий власний вектор буде досягнутий, якщо виконається умова $\|x^{k+1} - x^k\| \leq \varepsilon$, де ε – задана точність розрахунків. Збіжність методу спряжених напрямів слідує безпосередньо із збіжності методу простої ітерації (2.5).

Зауважимо, що значення

$$\alpha = \beta \frac{(x^k)^T Q^* x^{k+1}}{(x^k)^T Q^* x^k},$$

де $\beta \in (0,1]$, зберігає спряженість векторів x^k та x^{k+1} . Таким чином, β – регульований параметр, і швидкість збіжності розглянутого методу іноді залежить від вибору відповідного значення параметра β . Для скорочення обчислень у формулі для параметра α замінимо матрицю Q^* на обернену. Тоді

$$\alpha = \beta \frac{(x^k)^T (Q^*)^{-1} x^{k+1}}{(x^k)^T (Q^*)^{-1} x^k} = \beta \frac{\|x^{k+1}\|^2}{(x^k)^T x^{k+1}}.$$

Це дозволяє збільшити швидкість і точність обчислення власного значення (рекомендується вибирати $\beta = 0,5$).

Після того, як власний вектор x^* знайдений, мінімальне власне значення знаходимо за формулою

$$\lambda_{\min} = \frac{x^{*T} Q x^*}{x^{*T} x^*}.$$

Додатну напіввизначеність матриці Q визначаємо наступним чином: якщо $x^{*T} Q x^* \geq 0$, то матриця Q є додатно напіввизначеною; якщо $x^{*T} Q x^* < 0$, то матриця Q не є додатно напіввизначеною.

Збіжність розглянутого методу до власного вектора, відповідного мінімальному власному значенню, залежить від вибору початкового вектора. В якості початкового наближення x^0 достатньо взяти точку (вектор), яка не є ортогональною шуканому власному вектору. Цій умові задовольняє точка дотику еліпсоїда до грані описаного прямокутного паралелепіпеда

$$x^0 = \frac{(Q^*)^{-1}e}{\sqrt{a_{ii}^{-1}}},$$

де $e = (1, \dots, 1)$, а q_{ii}^{-1} – i -й діагональний елемент матриці $(Q^*)^{-1}$, значення i обирається з умови максимуму $\|x^0\|$.

2.1.2 Алгоритм методу спряжених напрямів

Вхідні дані: матриця Q розмірності $(n \times n)$, константа $\varepsilon > 0$, $\beta = 0,5$.

Вихідні дані: вектор-стовпчик x , що відповідає мінімального власному значенню матриці Q .

Крок 1. Обрати таке значення $r > 0$, що $r > \max \left\{ \sum_{i \neq j} |q_{ij}| - q_{ii}, \forall i, j \right\}$.

Крок 2. Знайти $Q^* = Q + rI$.

Крок 3. Обчислити початкову точку $x^0 = \frac{(Q^*)^{-1}e}{\sqrt{q_{ii}^{*-1}}}$, де $e = (1, \dots, 1)$, а q_{ii}^{*-1} –

i -й діагональний елемент матриці $(Q^*)^{-1}$, значення i обирається з умови максимуму $\|x^0\|$.

Крок 4. Обчислити $x^{k+1} = \frac{(Q^*)^{-1}x^k}{\sqrt{(x^k)^T (Q^*)^{-1} x^k}}$.

Крок 5. Обчислити $\alpha = \beta \frac{(x^k)^T Q^* x^{k+1}}{(x^k)^T Q^* x^k}$.

Крок 6. Обчислити $z^{k+1} = x^{k+1} - \alpha x^k$.

Крок 7. Якщо $\|x^{k+1} - x^k\| \geq \varepsilon$, прирівняти $x^{k+1} = z^{k+1}$ та перейти на крок 4.

Крок 8. Якщо $\|x^{k+1} - x^k\| < \varepsilon$, то отриманий x є розв'язком.

2.1.3 Приклад

Нехай маємо задачу:

$$\min\left\{x^T \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} x \mid \|x\|^2 = 1\right\}. \quad (2.6)$$

Її розв'язок дорівнює $(0, \pm 1)$. Розв'яжемо задачу (2.6) методом спряжених напрямів з точністю $\varepsilon = 1E-05$.

Розв'язок задачі (2.6) еквівалентний розв'язку наступної (до цільової функції додається доданок $2(\|x\|^2 - 1)$, який дорівнює нулю)

$$\min\left\{x^T \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} x + 2(\|x\|^2 - 1) \mid \|x\|^2 = 1\right\},$$

або

$$\min\left\{x^T \left(\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + 2I\right)x - 2 \mid \|x\|^2 = 1\right\}. \quad (2.7)$$

Так як константа $r = 2$ не впливає на значення точки мінімуму x , то її можна опустити. Тобто, розв'язок задачі (2.7) еквівалентний розв'язку наступної:

$$\min\left\{x^T \left(\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + 2I\right)x \mid \|x\|^2 = 1\right\}.$$

Отримали задачу

$$\min\left\{x^T \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} x \mid \|x\|^2 = 1\right\}, \quad (2.8)$$

яка відрізняється від задачі (2.6) тим, що матриця $Q^* = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$ в цільовій функції є додатно напіввизначеною. Очевидно, що розв'язком задачі (2.8) є

точка $(0, \pm 1)$, яка співпадає з розв'язком задачі (2.7) та розв'язом початкової задачі (2.6).

Згідно теореми 2.1, далі отримуємо задачу максимуму кулі на еліпсоїді:

$$\max\{\|x\|^2 \mid x^T \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} x = 1\}. \quad (2.9)$$

Оберемо початкову точку:

$$x^0 = \begin{pmatrix} 0,57735 \\ 1,73205 \end{pmatrix}.$$

Тоді

$$x^1 = \begin{pmatrix} 0,10911 \\ 0,98198 \end{pmatrix}, \alpha = 0,27672, z^1 = \begin{pmatrix} -0,05066 \\ 0,50268 \end{pmatrix}.$$

Прирівняємо $x^1 = z^1$ та будемо шукати наступне наближення. Здійснивши ще вісім ітерацій, отримаємо:

$$x^2 = \begin{pmatrix} -0,03353 \\ 0,99831 \end{pmatrix}, x^3 = \begin{pmatrix} 0,01109 \\ 0,99982 \end{pmatrix}, x^4 = \begin{pmatrix} -0,00369 \\ 0,99998 \end{pmatrix},$$

$$x^5 = \begin{pmatrix} 0,00123 \\ 0,99999 \end{pmatrix}, x^6 = \begin{pmatrix} -0,00041 \\ 0,99999 \end{pmatrix}, x^7 = \begin{pmatrix} 0,00013 \\ 0,99999 \end{pmatrix},$$

$$x^8 = \begin{pmatrix} -4,5E-05 \\ 0,99999 \end{pmatrix}, x^9 = \begin{pmatrix} 1,52E-06 \\ 1 \end{pmatrix}.$$

Отримана точка $x^9 = \begin{pmatrix} 1,52E-06 \\ 1 \end{pmatrix}$ є розв'язком задачі (2.9), задачі (2.8),

задачі (2.7) та початкової задачі (2.6).

2.1.4 Порівняльні чисельні експерименти

Засобами VBA Excel були розроблені програми методу QR, методу зворотної ітерації, методу ітерації з відношенням Релея і даної модифікації, що використовує спряжені напрями (відповідно QR, PI, PIR, PID), а також методу Якобі (RM) (див. п. 1.7). Експерименти проводилися для матриць різного розміру зі складною структурою на комп'ютері з двох-ядерним процесором Intel PentiumCore i5 з частотою 2,5 ГГц при точності обчислень 10^{-8} . Симетричні матриці перетворювалися до додатно визначених, для яких знаходився власний вектор, що відповідає мінімальному власному значенню. Максимальна кількість ітерацій для кожного методу обиралася рівною 50. У таблиці 2.1 наведено результати розрахунків власного вектору для складних матриць. Точність розрахунків оцінювалася за формулою $v = (Q - \lambda_{\min} I)x$ та приведено середнє значення компонент вектора v . Час розрахунків для всіх методів не указаний, так як він невеликий (порядка $10E-4$ сек.).

Таблиця 2.1 – Порівняльна таблиця методів для знаходження мінімального власного значення

	Розмірність матриці	Критерій	Методи				
			PID	PI	PIR	RM	QR
1	6×6	власне значення	1,00096	1,00140	1,00786	1,00105	1,00096
		точність розрахунків	6,374E-07	-1,8E-05	-0,00085	-0,00248	2,132E-10
2	11×11	власне значення	4,98902	4,99998	4,99964	4,99668	4,989020
		точність розрахунків	8,314E-07	-6,8E-07	0,000444	-6,8E-05	3,165E-10
3	7×7	власне значення	0,001259	0,001259	0,012935	55,97033	0,001259
		точність розрахунків	3,305E-12	4,4E-12	24,01784	39,69348	0,000216
4	9×9	власне значення	0,985013	0,985013	0,988491	0,985013	0,985013
		точність розрахунків	2,693E-16	1,22E-09	1,996E-03	2,93E-07	6,929E-11

Продовження таблиці 2.1

	Розмірність матриці	Критерій	Методи				
			PID	PI	PIR	RM	QR
5	10×10	власне значення	2,906279	2,906289	2,907956	2,907980	2,906279
		точність розрахунків	8,089E-07	2,310E-04	0,0051	0,00015	2,537E-09
6	11×11	власне значення	0,440913	0,441096	0,441465	0,441409	0,531315
		точність розрахунків	1,74E-04	1,91E-04	3,1E-05	2,1E-03	3,391E-11
7	5×5	власне значення	7,010653	7,010653	7,010653	7,010653	7,010653
		точність розрахунків	4,179E-09	1,32E-05	7,090E-06	4,135E-08	4,735E-09
8	5×5	власне значення	8,856101	8,856101	8,856101	8,856101	8,856101
		точність розрахунків	2,883E-13	6,43E-07	6,713E-05	3,366E-08	4,877E-11
9	10×10 (матриця Гільберта)	власне значення	4,751E-13	1,093E-13	3,953E-09	9,21E-04	1,093E-13
		точність розрахунків	2,211E-07	1,317E-18	2,184E-05	7,85E-06	1,308E-10

Перевага методу PID спостерігалася і в багатьох інших експериментах зі складними матрицями. Інколи інші методи знаходили мінімальне власне значення, але з більшою похибкою при визначенні відповідного власного вектора. Але не завжди метод спряжених напрямів знаходив кращий результат. Так, для матриці Гільберта розмірності 10 кращий результат у метода оберненої ітерації.

Таким чином, запропонована модифікація методу оберненої ітерації, що використовує спряжені напрями, дозволяє збільшити точність знаходження власного вектору матриці (за результатами чисельних експериментів). Це дозволить з великою точністю визначати додатну напіввизначеність матриць, що важливо для прискорення збіжності напіввизначеного симплекс-методу (похибки в обчисленні власного вектору можуть привести до передчасного завершення ітерацій напіввизначеного симплекс-методу).

2.2 Напіввизначений симплекс-метод

Розглянутий в підрозділі 1.2 прямо-двоїстий метод внутрішньої точки знайшов широке застосування в задачах напіввизначеної оптимізації, але на

відміну від лінійного програмування виникли проблеми при розв'язуванні задач великої розмірності, з лінійно-залежними матрицями та розривом двоїстості, при розв'язуванні задач з недопустимою початковою точкою та при встановленні пустоти та необмеженості розв'язків. Ці проблеми ефективно розв'язуються симплекс-методом для лінійного програмування. Тому узагальнення симплекс-методу для розв'язування задач напіввизначеної оптимізації є досить актуальним. Не дивлячись на значні зусилля в цьому напрямку, напіввизначений симплекс-метод [125] є першою ефективною реалізацією симплекс-методу.

У задачі напіввизначеного програмування потрібно знайти

$$\min\{C \bullet X \mid A_i \bullet X = b_i, X \succeq 0, i = 1, \dots, m\}, \quad (2.10)$$

де X – симетрична додатно напіввизначена матриця розмірності $(n \times n)$;

C та усі A_i – симетричні матриці $(n \times n)$;

b – вектор-стовпчик розмірності m ;

b_1, \dots, b_m – компоненти вектора b ;

$$C \bullet X = \sum \sum c_{ij} x_{ij}.$$

Теорема 2.3. Ранг матриці X задачі (2.10) не перевищує числа m .

Доведення. Використаємо нерівність $\text{rank } X \leq \left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor$, доведену в

роботі [26 с. 270]. Покажемо, що $\text{rank } X \leq m$. Маємо:

$$\frac{\sqrt{8m+1}-1}{2} \leq m,$$

$$\sqrt{8m+1} \leq 2m+1,$$

$$8m+1 \leq 4m^2 + 4m + 1,$$

$$4m(m-1) \geq 0.$$

З того, що остання нерівність завжди виконується для цілого m , випливає, що виконується нерівність $\frac{\sqrt{8m+1}-1}{2} \leq m$, тобто ранг матриці X задачі (2.10) не перевищує числа m . Теорему доведено.

Безпосередньо з теореми 2.3 випливає, що зменшення кількості обмежень призводить до зменшення рангу матриці X , а додавання обмежень не завжди призводить до збільшення рангу матриці X та інколи навіть зменшує ранг матриці (див. п. 3.2).

2.2.1 Перетворення задачі SDP до задачі лінійного програмування

Розглянемо пряму задачу напіввизначеного програмування (2.10), де необхідно знайти додатно напіввизначену матрицю X^* . Відомо, що будь-яка додатно напіввизначена матриця може бути розкладена на суму матриць рангу одиниця [168 с. 542], які можна представити у вигляді xx^T . Таким чином, матриці рангу одиниця визначаються векторами $x^j = (x_1^j, \dots, x_n^j)$. Ці матриці являють собою крайні промені (твірні) конуса додатно напіввизначених матриць S_n^+ , а їх опукла комбінація визначає S_n^+ . Число таких твірних нескінченне, але існує їх скінченний набір, який утворює конус K^* (конус K^* – це конус, який містить розв’язок задачі (2.10)). В напіввизначеному симплекс-методі початковий напіввизначений конус вибирається у вигляді багатогранного конусу, що визначається набором його твірних. Оберемо початковий набір твірних, які визначаються векторами x_j з компонентами рівними $-1, 0, 1$, причому тільки дві або одна компоненти цих векторів відмінні від нуля. Цей набір векторів є достатнім, але не необхідним, так як доведення збіжності напіввизначеного симплекс-методу не залежить від початкової кількості векторів. Ця кількість тільки впливає на число ітерацій методу.

Лема 2.1. Число матриць рангу одиниця, що утворені векторами x_j , дорівнює $n(n+1)/2$.

Доведення. Перестановка ненульових компонент в векторі x_j породжує

одну і ту ж матрицю рангу одиниця. Вектори x_j будемо отримувати шляхом зміщення ненульових компонент з фіксованою кількістю нулів між ними. Отримуємо наступну кількість матриць: $(n-1)$ матриць без нулів між ненульовими компонентами, $(n-2)$ матриць з одним нулем між ненульовими компонентами, ..., 1 матрицю з $(n-2)$ нулями між ненульовими компонентами. Загальна кількість цих матриць дорівнює

$$(n-1+1)\frac{(n-1)}{2}.$$

Кількість матриць, утворених з векторів x_j з однією компонентою, відмінною від нуля, дорівнює n . Тоді загальна кількість матриць рангу одиниця, які визначаються векторами x_j з компонентами рівними $-1, 0, 1$, причому тільки дві або одна компоненти цих векторів відмінні від нуля, дорівнює

$$\frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}.$$

Лема доведена.

Позначимо ці матриці через $X_j = x_j x_j^T$, а відповідний багатогранний конус через K_0 . Будемо шукати розв'язок задачі (2.10) у вигляді $X = \sum_j \alpha_j X_j$, де число доданків дорівнює $n(n+1)/2$ та $\alpha \geq 0$. Тоді задача (2.10) перетвориться до вигляду

$$\min\{C \bullet \sum_j \alpha_j X_j \mid A_i \bullet \sum_j \alpha_j X_j = b_i, i=1, \dots, m, \alpha \geq 0\},$$

або

$$\min\{\sum_j \alpha_j C \bullet X_j \mid \sum_j \alpha_j A_i \bullet X_j = b_i, i=1, \dots, m, \alpha \geq 0\}. \quad (2.11)$$

Розв'язок задачі (2.11) α^* визначить додатно напіввизначену матрицю $X = \sum_j \alpha_j^* X_j$, яка буде розв'язком задачі (2.10), якщо $X^* \subseteq K_0$ (це положення доводиться у теоремі 2.4 далі). У протилежному випадку, конус K_0 необхідно збільшити (див. теорему 2.5 нижче).

Теорема 2.4. *Якщо багатогранний конус, що визначається X_j , містить розв'язок задачі SDP (2.10), то оптимальне значення α задачі ЛП (2.11) визначить розв'язок задачі SDP.*

Доведення. Так як розв'язок X^* задачі SDP належить багатогранному конусу, то існують такі $\alpha_j \geq 0$, що

$$X^* = \sum_j \alpha_j X_j.$$

Помножимо обидві частини цієї рівності на C , отримаємо

$$C \bullet X^* = C \bullet \sum_j \alpha_j X_j. \quad (2.12)$$

Покажемо, що не існує такого розв'язку α^* задачі ЛП, для якого виконується нерівність

$$C \bullet \sum_j \alpha_j^* X_j < C \bullet \sum_j \alpha_j X_j. \quad (2.13)$$

Допустима множина задачі ЛП є підмножиною допустимої множини задачі SDP, звідки випливає, що

$$C \bullet X^* \leq C \bullet \sum_j \alpha_j^* X_j. \quad (2.14)$$

Віднімемо з (2.13) рівність (2.12), отримаємо

$$C \bullet X^* > C \bullet \sum_j \alpha_j^* X_j,$$

що суперечить (2.14). Таким чином, у формулі (2.13) може бути тільки рівність та

$$C \bullet X^* = C \bullet \sum_j \alpha_j^* X_j.$$

Теорема доведена.

Лема 2.2. *Якщо задача SDP має розв'язок та її допустима множина є обмежена, тоді для довільного вектора x існує такий індекс i , що нерівність $A_i x x^T > 0$ виконується.*

Доведення. Припустимо обернене, що для всіх i $A_i x x^T \leq 0$. Тоді існує така матриця C , що відповідна оцінка в рядку цільової функції буде від'ємною. При цій умові розв'язок задачі SDP буде необмежений. Отримано протиріччя, яке доводить лему.

Зауваження 2.1. В задачах лінійного програмування можливі ситуації, коли на наступній ітерації значення цільової функції не змінюється. Це трапляється тоді, коли поточний розв'язок є виродженим. Якщо через декілька ітерацій симплекс-метод повертається до попереднього базису і при цьому всі розв'язки є виродженими, то виникає зациклення. Розглянута у п.1.6 теорема Бленда дає простий алгоритм виходу з цього зациклення, тому виродженість у симплекс-методі можна розглядати як локальне явище. В напіввизначеному симплекс-методі питання виродженості вирішується при розв'язанні задачі лінійного програмування (2.11), яка розв'язується звичайним симплекс-методом. Крім того, для розв'язування задачі лінійного програмування (2.11) можна використовувати прямо-двоїстий метод внутрішньої точки, для якого

виродженість не є проблемою (значення цільової функції спадає).

Теорема 2.5. *Нехай задача SDP має розв'язок та D_0 – її допустима множина. Тоді додавання в її матрицю обмежень нового стовпця утворює множину D_1 , таку що $D_0 \subseteq D_1$.*

Доведення. Перетворимо рядок цільової функції згідно з алгоритмом симплекс-методу. Якщо отриманий коефіцієнт цільової функції, що відповідає новому стовпцю, є від'ємним, то наступна ітерація симплекс-методу зменшить значення цільової функції. Згідно з лемою 2.2 це зменшення буде скінченним. Якщо коефіцієнт цільової функції невід'ємний, то знайдеться матриця C , для якої він буде від'ємним. Таким чином, додавання нового стовпця приводить до нових недопустимих точок. В той же час, всі допустимі точки початкової задачі зберігаються. Тобто, якщо поточний розв'язок не вироджений, то $D_0 \subset D_1$. Якщо поточний розв'язок вироджений, то $D_0 = D_1$. Теорема доведена.

Задача (2.11) є задачею лінійного програмування, яку будемо розв'язувати симплекс-методом. У симплекс-методі на кожній ітерації знаходиться базисний розв'язок та перетворюється рядок цільової функції так, щоб при базисних змінних коефіцієнти цільової функції (оцінки) були рівними нулю. В новий базисний розв'язок включається стовпець матриці обмежень з мінімальним значенням коефіцієнта перетвореної цільової функції. Цей базисний розв'язок зменшує значення цільової функції, якщо оцінка в рядку цільової функції від'ємна. Для напіввизначеного симплекс-метода новий стовпець матриці обмежень потрібно побудувати.

2.2.2 Пошук коефіцієнта цільової функції

Новий стовпець матриці обмежень будемо визначати матрицею рангу одиниця. Позначимо цей стовпець через $x_k x_k^T$ та знайдемо для нього коефіцієнт цільової функції.

Симплекс-таблиця має такий вигляд

$$\begin{array}{cccccc|c}
 A_1 \bullet x_1 x_1^T & A_1 \bullet x_2 x_2^T & \dots & A_1 \bullet x_m x_m^T & \dots & A_1 \bullet x_k x_k^T & b_1 \\
 A_2 \bullet x_1 x_1^T & A_2 \bullet x_2 x_2^T & \dots & A_2 \bullet x_m x_m^T & \dots & A_2 \bullet x_k x_k^T & b_2 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 A_m \bullet x_1 x_1^T & A_m \bullet x_2 x_2^T & \dots & A_m \bullet x_m x_m^T & \dots & A_m \bullet x_k x_k^T & b_m \\
 \hline
 C \bullet x_1 x_1^T & C \bullet x_2 x_2^T & \dots & C \bullet x_m x_m^T & \dots & C \bullet x_k x_k^T &
 \end{array}$$

Для шуканого стовпця $x_k x_k^T$ оцінка в перетвореному рядку цільової функції повинна бути мінімальною. Знайдемо цю оцінку.

Нехай перші m стовпців симплекс-таблиці входять в базис. Позначимо через B – матрицю базисних елементів оптимального розв’язку задачі (2.11), через B^{-1} – обернену матрицю до матриці базисних елементів, а через b_{ij}^{-1} – елементи матриці B^{-1} . Тоді для того, щоб отримати одиничну матрицю, помножимо матрицю B^{-1} на симплекс-таблицю. Отримаємо:

$$\begin{array}{cccccc|c}
 1 & 0 & \dots & 0 & \sum_{j=1}^m b_{1j}^{-1} A_j \bullet x_{m+1} x_{m+1}^T & \dots & \sum_{j=1}^m b_{1j}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{1j}^{-1} b_j \\
 0 & 1 & \dots & 0 & \sum_{j=1}^m b_{2j}^{-1} A_j \bullet x_{m+1} x_{m+1}^T & \dots & \sum_{j=1}^m b_{2j}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{2j}^{-1} b_j \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & 1 & \sum_{j=1}^m b_{mj}^{-1} A_j \bullet x_{m+1} x_{m+1}^T & \dots & \sum_{j=1}^m b_{mj}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{mj}^{-1} b_j \\
 \hline
 C \bullet x_1 x_1^T & C \bullet x_2 x_2^T & \dots & C \bullet x_m x_m^T & C \bullet x_{m+1} x_{m+1}^T & \dots & C \bullet x_k x_k^T &
 \end{array}$$

Таким чином, елементи нового k -го стовпця матриці обмежень задачі (2.11) будуть дорівнювати

$$B^{-1}(A_1 \bullet x_k x_k^T, A_2 \bullet x_k x_k^T, \dots, A_m \bullet x_k x_k^T),$$

а рядок цільової функції перетворимо наступним чином: щоб зробити рядок

цільової функції нульовим напроти стовпців, які входять в базис, необхідно перший рядок симплекс-таблиці помножити на $C \bullet x_1 x_1^T$, другий рядок – на $C \bullet x_2 x_2^T$, ..., останній рядок – на $C \bullet x_m x_m^T$, та від рядка цільової функції відняти отримані значення. Отримаємо:

$$\begin{array}{cccccc|c}
 1 & 0 & \dots & 0 & \dots & \sum_{j=1}^m b_{1j}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{1j}^{-1} b_j \\
 0 & 1 & \dots & 0 & \dots & \sum_{j=1}^m b_{2j}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{2j}^{-1} b_j \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & 1 & \dots & \sum_{j=1}^m b_{mj}^{-1} A_j \bullet x_k x_k^T & \sum_{j=1}^m b_{mj}^{-1} b_j \\
 \hline
 0 & 0 & \dots & 0 & \dots & C \bullet x_k x_k^T - \sum_{i=1}^m C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j \bullet x_k x_k^T &
 \end{array}$$

Таким чином, рядок цільової функції потрібно перетворити за формулою

$$C \bullet x_k x_k^T - \sum_i C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j \bullet x_k x_k^T,$$

або

$$(C - \sum_i C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j) \bullet x_k x_k^T, \quad (2.15)$$

де сумування проводиться по всім базисним стовпцям матриці обмежень задачі (2.11).

Покажемо, що оцінка (2.15) співпадає з тою, що отримується за допомогою методу генерації стовпців [144].

Розглянемо задачу лінійної оптимізації в загальному вигляді:

$$\min \left\{ \sum_{j=1}^n c_j x_j \mid \sum_{j=1}^n p_{ij} x_j = b_i, x_j \geq 0, j=1, \dots, n, i=1, \dots, m \right\},$$

де p_j та b – m -вимірні вектор-стовпчики з компонентами (p_{1j}, \dots, p_{mj}) та (b_1, \dots, b_m) відповідно;

c та x – n -вимірні вектор-стовпчики з компонентами (c_1, \dots, c_n) та (x_1, \dots, x_n) відповідно.

Для того, щоб визначити можливість покращення першого допустимого розв'язку, необхідно обчислити характеристичну різницю:

$$c'_j = c_j - c_B B^{-1} p_j,$$

де p_j – небазисний вектор;

c_j – коефіцієнт цільової функції при небазисному векторі p_j ;

c_B – вектор коефіцієнтів цільової функції при базисному розв'язку;

B^{-1} – матриця, обернена до матриці базису.

Якщо $\min_j c_j = \bar{c}_s < 0$, то початковий розв'язок можна покращити шляхом включення у базис вектора x_s та виконання перетворень (ведучої операції).

Для нашої задачі (2.11) припустимо, що перші m стовпців симплекс-таблиці входять в базис, тоді $c_B = (C \bullet x_1 x_1^T, C \bullet x_2 x_2^T, \dots, C \bullet x_m x_m^T)$, $c_j = C \bullet x_k x_k^T$, $p_j = A_i \bullet x_k x_k^T$, звідки маємо:

$$c'_j = C \bullet x_k x_k^T - (C \bullet x_1 x_1^T, \dots, C \bullet x_m x_m^T) \begin{pmatrix} b_{11}^{-1} & \dots & b_{1m}^{-1} \\ \vdots & & \vdots \\ b_{m1}^{-1} & \dots & b_{mm}^{-1} \end{pmatrix} \begin{pmatrix} A_1 \bullet x_k x_k^T \\ \vdots \\ A_m \bullet x_k x_k^T \end{pmatrix},$$

$$c'_j = C \bullet x_k x_k^T - (C \bullet x_1 x_1^T b_{11}^{-1} + \dots + C \bullet x_m x_m^T b_{m1}^{-1}, \dots, C \bullet x_1 x_1^T b_{1m}^{-1} + \dots + C \bullet x_m x_m^T b_{mm}^{-1}) \begin{pmatrix} A_1 \bullet x_k x_k^T \\ \vdots \\ A_m \bullet x_k x_k^T \end{pmatrix},$$

$$c'_j = C \bullet x_k x_k^T - \left(\sum_{i=1}^m C \bullet x_i x_i^T b_{i1}^{-1}, \dots, \sum_{i=1}^m C \bullet x_i x_i^T b_{im}^{-1} \right) \begin{pmatrix} A_1 \bullet x_k x_k^T \\ \vdots \\ A_m \bullet x_k x_k^T \end{pmatrix},$$

$$c'_j = C \bullet x_k x_k^T - \left(\sum_{i=1}^m C \bullet x_i x_i^T b_{i1}^{-1} A_1 \bullet x_k x_k^T + \dots + \sum_{i=1}^m C \bullet x_i x_i^T b_{im}^{-1} A_m \bullet x_k x_k^T \right),$$

$$c'_j = C \bullet x_k x_k^T - \sum_{i=1}^m C \bullet x_i x_i^T \left(\sum_{i=1}^m b_{i1}^{-1} A_1 \bullet x_k x_k^T + \dots + \sum_{i=1}^m b_{im}^{-1} A_m \bullet x_k x_k^T \right),$$

$$c'_j = C \bullet x_k x_k^T - \sum_{i=1}^m C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j \bullet x_k x_k^T \quad (2.16)$$

Отже, коефіцієнт цільової функції (2.15), відповідний стовпцю, який буде додано у напіввизначеному симплекс-методі, співпадає з оцінкою (2.16), що отримується за допомогою метода генерації стовпців [144].

2.2.3 Розв'язування квадратичної підзадачі

Позначимо $Q = C - \sum_i C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j$ (матриця Q буде симетричною),

тоді, згідно [18], можна записати

$$Q \bullet x_k x_k^T = x_k^T Q x_k.$$

Цей вираз $x_k^T Q x_k$ і буде необхідною оцінкою та необхідно знайти таке x_k , щоб оцінка $x_k^T Q x_k$ була мінімальною. Якщо $x_k^T Q x_k < 0$, то введення k -го стовпця в

базис призведе до зменшення цільової функції задачі (2.11) [113]. Якщо ж матриця Q – додатно напіввизначена, то $x^T Q x \geq 0, \forall x$ і значення цільової функції задачі (2.11) не може бути зменшено, тоді поточний розв’язок є оптимальним для задачі (2.10) (див. теорему 2.6). Для знаходження x_k розв’яжемо задачу квадратичної оптимізації

$$\min\{x^T Q x \mid \|x\|^2 = 1\}. \quad (2.17)$$

методом спряжених напрямів, описаним у п. 2.1.

Якщо x^* – розв’язок задачі (2.17), то матриця Q – додатно напіввизначена при умові $x^{*T} Q x^* \geq 0$. У цьому випадку задача (2.10) розв’язана, у протилежному випадку пошук розв’язку задачі (2.11) симплекс-методом буде продовжено. Це твердження впливає з теореми 2.6 (див. п. 2.3).

Якщо допустима множина задачі (2.11) пуста, то для знаходження начального базису використовуємо метод штучного базису (стовпці штучного базису J будуть визначатися матрицями довільного рангу). Мінімізуємо суму змінних $\alpha_k, k \in J$ для стовпців штучної матриці, вводячи нові стовпці в матрицю обмежень задачі (2.11). Якщо задача (2.10) має розв’язок, то усі $\alpha_k, k \in J$ стануть рівними нулю і буде знайдений початковий базис, який складається з матриць рангу одиниця.

2.2.4 Алгоритм напіввизначеного симплекс-методу

Вхідні дані: матриці C та A_i розмірності $(n \times n)$, вектор-стовпчик b розмірності m , константа $\varepsilon > 0$.

Вихідні дані: додатно напіввизначена матриця X розмірності $(n \times n)$.

Крок 1. Обрати початковий набір матриць рангу одиниця $X_j = x_j x_j^T$, які визначаються векторами x_j з компонентами рівними $-1, 0, 1$, причому тільки дві або одна компоненти цих векторів відмінні від нуля.

Крок 2. Розв'язати задачу

$$\min\{\sum_j \alpha_j C \bullet X_j \mid \sum_j \alpha_j A_i \bullet X_j = b_i, i=1, \dots, m, \alpha_j \geq 0\} \quad (2.18)$$

симплекс-методом для лінійного програмування відносно α_j . Отримати розв'язок $X^* = \sum_j \alpha_j^* X_j$, де α_j^* – оптимальний розв'язок задачі (2.18).

Крок 3. Знайти $Q = C - \sum_i C \bullet x_i x_i^T \sum_{j=1}^m b_{ij}^{-1} A_j$, де b_{ij}^{-1} – елементи матриці B^{-1} ,

яка є оберненою матрицею до матриці базисних елементів.

Крок 4. Розв'язати задачу

$$\min\{x^T Q x \mid \|x\|^2 = 1\}$$

методом спряжених напрямів, описаним у п. 2.1, отримати розв'язок x^* .

Крок 5. Якщо $x^{*T} Q x^* < 0$, то прийняти $X_j = x_j x_j^T + x^* x^{*T}$ та перейти на крок 2.

Крок 6. Якщо $x^{*T} Q x^* \geq 0$, то матриця X^* , отримана на кроці 2, є шуканим розв'язком.

2.2.5 Приклад

Розглянемо приклад

$$C = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}.$$

Початкова матриця для задачі (2.11) дорівнює (однакові стовпці програмою видаляються)

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 2 & -2 & -1 & 3 \end{pmatrix}$$

і додані стовпці (на кожній ітерації додається один стовпець)

$$\begin{pmatrix} 4,59\text{E-}02 & 0,561676 & 0,429775 & 7,88\text{E-}02 & 0,319177 \\ 4,59\text{E-}02 & 0,124838 & 0,122866 & 1,31\text{E-}02 & 5,59\text{E-}02 \\ 1 & -0,21611 & -1,22\text{E-}02 & 0,972314 & 0,357683 \end{pmatrix}$$

Таким чином, за 6 ітерацій був знайдений розв'язок

$$\begin{pmatrix} 4 & -2,61987 & -2,913056 \\ -2,61987 & 6,826113 & -0,210867 \\ -2,91306 & -0,21087 & 3 \end{pmatrix}$$

зі значенням цільової функції $C \bullet X = -25,313702$. Метод внутрішньої точки знайшов розв'язок

$$\begin{pmatrix} 3,99999727 & -2,62082 & -2,91425 \\ -2,6208194 & 6,828507 & -0,20752 \\ -2,9142529 & -0,20752 & 2,999999 \end{pmatrix}$$

за 22 ітерації з невеликою похибкою в обмеженнях і значенням цільової функції $C \bullet X = -25,313697$.

2.2.6 Переваги напіввизначеного симплекс-методу

Розглянутий симплекс-метод має ряд переваг перед методами внутрішньої точки. Перерахуємо ці переваги.

1. Розмірність задачі, яка розв'язується напіввизначеним симплекс-

методом, дорівнює $q+k$, де $q \geq m$ – початкова кількість матриць рангу одиниця (для даної реалізації симплекс-методу $q \leq \frac{n(n+1)}{2}$), k – число ітерацій методу. Розмірність задачі, яка розв’язується методом внутрішньої точки, дорівнює $m+(n+1)n$, де m – розмірність вектору змінних у двоїстої задачі, $(n+1)n$ – кількість змінних (компонент) симетричних матриць X та Z . При перетворенні задачі SDP до канонічного виду вводяться вільні змінні, які для методу внутрішньої точки потребують відповідного збільшення усіх матриць обмежень. Для напіввизначеного симплекс-методу таке збільшення непотрібно.

2. Ще більша різниця у числі обмежень. Число обмежень задачі при розв’язуванні симплекс-методом дорівнює m . Число обмежень задачі при розв’язуванні методом внутрішньої точки дорівнює $m+n(n+1)/2+n^2$, де m – кількість обмежень прямої задачі, $n(n+1)/2$ – кількість обмежень двоїстої задачі, n^2 – кількість обмежень, які отримують з умови рівності цільових функцій прямої та двоїстої задач (враховуючи те, що добуток симетричних матриць не є симетричною матрицею).

3. Область задач, які можуть бути розв’язані напіввизначеним симплекс-методом, є ширшою, тому що не потребується рівність цільових функцій прямої та двоїстої задач, так як напіввизначеним симплекс-методом розв’язується тільки пряма задача. Крім того напіввизначений симплекс-метод не потребує лінійної незалежності матриць обмежень.

4. Напіввизначений симплекс-метод є нечутливим до вибору початкової точки (використовується метод штучного базису), в той час як збіжність методів внутрішньої точки залежить від вибору початкової точки при розв’язуванні задачі.

5. Точність розв’язків, отриманих напіввизначеним симплекс-методом, є значно вищою, так як в методі внутрішньої точки використовується штраф, що є наближеним перетворенням.

6. Метод внутрішньої точки містить два параметри, від значень яких залежить його збіжність, в той же час напіввизначений симплекс-метод не

містить будь-яких параметрів.

2.3 Збіжність напіввизначеного симплекс-методу для напіввизначеної оптимізації

Звичайний симплекс-метод для лінійного програмування збігається за скінченне число ітерацій. В розглянутому напіввизначеному симплекс-методі можлива нескінченна кількість ітерацій. Покажемо, що ця послідовність ітерацій збігається до розв'язку задачі напіввизначеної оптимізації.

Теорема 2.6. *Якщо існує розв'язок задачі (2.10) з меншим значенням цільової функції, ніж у точці $\sum_j \alpha_j^* X_j$ (α^* – розв'язок задачі (2.11)), то тоді існує матриця X_k рангу одиниця, така, що для розширеної задачі (2.11) справедлива нерівність $C \bullet (\sum_j \alpha_j X_j + \alpha_k X_k) < C \bullet \sum_j \alpha_j^* X_j$.*

Доведення. Нехай X^* – розв'язок задачі (2.10), тоді

$$X^* = \sum_i X_i^*,$$

де X_i^* – матриці рангу одиниця. Додамо до обмежень задачі (2.11) нові стовпці, що містять матриці X_i^*

$$\alpha_1 A_1 \bullet X_1 + \dots + \alpha_m A_1 \bullet X_m + \dots + \alpha_k A_1 \bullet X_1^* + \dots + \alpha_r A_1 \bullet X_r^* = b_1,$$

$$\dots \dots \dots$$

$$\alpha_1 A_m \bullet X_1 + \dots + \alpha_m A_m \bullet X_m + \dots + \alpha_k A_m \bullet X_1^* + \dots + \alpha_r A_m \bullet X_r^* = b_m.$$

Розв'язок задачі (2.11) при цих обмеженнях визначить матрицю X^* , а для цього, принаймні, одна оцінка в перетвореному рядку цільової функції, відповідна доданим стовпцям, буде від'ємною ($Q \bullet X_i^* < 0$), так як, в протилежному випадку, поточний розв'язок задачі (2.11) буде оптимальним.

Додавання одного з таких стовпців з від'ємною оцінкою в матриці обмежень задачі (2.11) призведе до спадання її цільової функції (див. теорему 4 в [113 с. 101] та зауваження 2.1), що доводить теорему.

Теорема 2.6 стверджує, що спадання цільової функції задачі (2.11), при додаванні нових стовпців в обмеження задачі, буде до тих пір, поки опукла оболонка матриць X_j не буде містити розв'язок X^* , але тоді розв'язок задачі (2.11) однозначно визначить розв'язок задачі (2.10).

Для доведення збіжності розглянутого симплекс-методу необхідно наступне твердження.

Теорема 2.7. *Гранична точка послідовності $\{x^k\}$ належить ε -околу точки мінімуму x^* неперервної функції $f(x)$ на компактній допустимій множині, якщо $f(x)$ – обмежена на цій множині і для довільного $\varepsilon > 0$ (ε – точність обчислень), значення $f(x)$ спадає, а поза ε -околом точки x^* справедлива нерівність $f(x^{k+1}) < f(x^k) - \delta$, $\delta > \varepsilon$, $\forall k$.*

Доведення. Послідовність $f(x^k)$ спадає і обмежена знизу, тоді вона сходиться до деякого значення $f(x^m)$, що означає $f(x^k) - f(x^m) < \varepsilon$, $\forall k < +\infty$. Якщо точка x^m не належить ε -околу точки x^* , то $f(x^{k+1}) < f(x^k) - \delta$. З останніх двох нерівностей випливає $f(x^{k+1}) < f(x^m)$, але тоді x^m не може бути граничною точкою. Таким чином, x^m може належати тільки ε -околу точки x^* . Теорема доведена.

Безпосередньо з теорем 2.6, 2.7 випливає збіжність напіввизначеного симплекс-методу.

Теорема 2.8. *Нехай знайдений розв'язок задачі (2.11), який визначає матрицю X^k і для якого виконується умова $C \bullet X^k - C \bullet X^* > \varepsilon > 0$ та $B(\varepsilon)$ – ε -оکیل розв'язку X^* задачі (2.10). Тоді гранична точка X^∞ , яка визначається послідовністю розв'язків задач (2.11), належить $B(\varepsilon)$.*

Доведення. Достатньо показати, що $C \bullet X^{k+1} < C \bullet X^k - \delta$ поза околом

$B(\varepsilon)$. З теореми 4 [113 с. 101] слідує, що

$$C \bullet X^{k+1} - C \bullet X^k = x_k^T Q x_k \frac{\bar{b}_r}{\bar{a}_{rk}} \leq 0,$$

де r – провідний рядок симплекс-метода;

x_k – розв’язок задачі (2.17).

Оберемо

$$\delta = \varepsilon - x_k^T Q x_k \frac{\bar{b}_r}{\bar{a}_{rk}} > \varepsilon, \text{ де } x_k^T Q x_k < 0,$$

тоді, якщо $\bar{b}_r > 0$ (розв’язок задачі (2.11) не вироджений), то умови теореми 2.7 виконуються. Якщо припустити, що $\bar{b}_r = 0, \forall k$, то значення цільових функцій задачі (2.11) при додаванні нових стовпців у матрицю обмежень не будуть спадати, що суперечить умовам теореми 2.6. Таким чином, $X^\infty \in B(\varepsilon)$. Теорему доведено.

2.4 Чисельні експерименти з напіввизначеним симплекс-методом

Розроблено програми напіввизначеного симплекс-методу і методу внутрішньої точки ІМР (див. п. 4.4).

Порівняльні чисельні експерименти проводилися для задач SDP (2.10) різної розмірності. Напіввизначений симплекс-метод не вимагає початкової допустимої точки (матриці). У програмі ІМР в якості початкової використовувалася одинична матриця. За критерії визначення ефективності методів бралися точність і час розв’язування, а також точність виконання обмежень. Експерименти показали переваги напіввизначеного симплекс-методу як за швидкістю розв’язування задач, так і за точністю виконання обмежень.

Порівняльні експерименти проводилися на комп'ютері з двоядерним процесором Intel Core i5 з частотою 2.5 ГГц. При точності обчислень 10^{-8} метод внутрішньої точки іноді давав похибку в обмеженнях порядку 0.001, а напіввизначений симплекс-метод знаходив розв'язки, які задовольняли обмеженням з машинною точністю.

Потім були проведені експерименти з існуючими програмними пакетами. Результати цих експериментів наведено в табл. 2.2.

Задачі напіввизначеної оптимізації розв'язувалися за допомогою пакетів CSDP, PENSDP, SDPA, SDPT 3, SEDUMI з використанням NEOS-сервера, для їх розв'язання за допомогою напіввизначеного симплекс-методу була розроблена програма на VBA для Excel. Також на VBA для Excel була розроблена програма розглянутого методу внутрішньої точки [82].

Таким чином, з результатів, представлених у табл. 2.2, можна зробити висновок, що напіввизначений симплекс-метод не поступається методам внутрішньої точки, навіть при тому, що його програмна реалізація не є оптимальною.

Таблиця 2.2 – Чисельні експерименти

№ П/ П	n	m	Критерій	Програмний пакет					
				Симплекс-метод	CSDP	PENSDP	SDPA	SDPT3	SEDUMI
1	10	10	розв'язок	-198,97	-199,03	-199,03	-199,03	-199,03	-199,03
			час	6,45e-02	6,65e-03	2e-02	4,51e-03	1,3	8e-01
2	11	14	розв'язок	-74,97	-75,02	-75,02	-75,02	-75,02	-75,02
			час	2,29e-01	9,24e-03	4e-02	8,51e-03	1,2	8,3e-01
3	20	3	розв'язок	-873,3804	-873,3806	-873,3806	-873,3806	-873,3806	-873,3806
			час	3,93e-02	1,073e-02	4e-02	1,26e-01	1,5	8,6e-01
4	30	1	розв'язок	-5852,52	-5852,52	-5852,52	-5852,52	-5852,52	-5852,52
			час	2,23e-02	1,41e-02	5e-02	9,84e-03	1,3	6,9e-01

Продовження таблиці 2.2

№ П/ П	n	m	Критерій	Програмний пакет					
				Симплекс-метод	CSDP	PENSDP	SDPA	SDPT3	SEDUMI
5	50	2	розв'язок	-3675,12	-3675,12	-3675,12	-3675,12	-3675,12	-3675,12
			час	1,71e-01	5,06e-02	1,4e-01	5,57e-02	1,7	1,06
6	100	2	розв'язок	-57408,05259	-57408,053	-57408,053	-57408,053	-57408,053	-57408,053
			час	5,6e-01	3,31e-01	6,2e-01	2,53e-01	3,1	3,63
7	25	4	розв'язок	-112232,2602	-112232,26	-112232,26	-112232,26	-112232,26	-112232,26
			час	1,48e-01	1,49e-02	8e-02	1,38e-02	1,5	9e-01
8	13	6	розв'язок	-9,194	-8,99	-9	-8,99	-8,99	-8,99
			час	5,79e-03	1,83e-02	3e-02	3,3e-02	1,6	7,7e-01

Розглянемо наступний приклад. Нехай маємо семивимірну задачу напіввизначеної оптимізації з вісьмома обмеженнями (дані наведені в таблиці Б.1). Ця задача була розв'язана напіввизначеним симплекс-методом. Отримали розв'язок

$$\begin{pmatrix} 1 & 15,786 & -0,977 & 1E-15 & 0,959 & 0,0666 & 0,2017 \\ 15,786 & 252,58 & -17,54 & -11,34 & 15,011 & 1,386 & 3,6641 \\ -0,977 & -17,54 & 5,557 & 5,2255 & -1,085 & -0,087 & -0,229 \\ 1E-15 & -11,34 & 5,2255 & 89,705 & 0,5423 & -0,961 & -3,924 \\ 0,959 & 15,011 & -1,085 & 0,5423 & 0,959 & 0,0238 & 0,1135 \\ 0,0666 & 1,386 & -0,087 & -0,961 & 0,0238 & 0,0666 & 0,0937 \\ 0,2017 & 3,6641 & -0,229 & -3,924 & 0,1135 & 0,0937 & 0,4033 \end{pmatrix}$$

та значення цільової функції $-25,915$. Усі обмеження були виконані.

Прямо-двоїстий метод внутрішньої точки не може знайти оптимальний розв'язок цієї задачі.

Інші приклади, де симплекс-метод знаходить розв'язок задачі

напіввизначеної оптимізації, а прямо-двоїсті методи внутрішньої точки знаходять недопустимий розв'язок, наведено у додатку А.

2.5 Висновки розділу

Основними результатами розділу є:

- уперше використано метод спряжених напрямів у процедурі оберненої ітерації для визначення додатної напіввизначеності матриць (шляхом знаходження власного вектора, відповідного мінімальному власному значенню), що за результатами чисельних експериментів дозволило підвищити точність розрахунків та прискорити збіжність до власного вектора матриці;

- для розв'язування задач напіввизначеної оптимізації удосконалено напіввизначений симплекс-метод та прискорено його збіжність шляхом використання процедури оберненої ітерації та методу спряжених напрямів;

- уперше строго доведено збіжність напіввизначеного симплекс-методу за допомогою теорем і лем;

- визначено теоретичні переваги напіввизначеного симплекс-методу над прямо-двоїстими методами внутрішньої точки;

- наведено приклади задач, для яких напіввизначений симплекс-метод знаходить розв'язки, а метод внутрішньої точки не знаходить;

- проведено порівняння результатів програми, яка реалізує напіввизначений симплекс-метод, з відомими програмними пакетами, які реалізують методи внутрішніх точок. Ці експерименти показали, що дана програмна реалізація напіввизначеного симплекс-методу не поступається популярним програмним пакетам методів внутрішньої точки навіть при тому, що його програмна реалізація не є оптимальною.

Основні результати розділу опубліковано в [77, 129, 131, 132, 134, 135, 155].

РОЗДІЛ 3

SDP ДЛЯ ЗАГАЛЬНИХ КВАДРАТИЧНИХ ЗАДАЧ

У цьому розділі описується використання напіввизначеної оптимізації для розв'язування загальних квадратичних задач, задачі максимізації норми вектора на перетині еліпсоїдів і квадратичних задач з булевими змінними, а також наводяться чисельні експерименти з використанням розроблених програм з критичним аналізом результатів. Для цих задач пропонується та перевіряється на практиці нова процедура розв'язання, яка ґрунтується на використанні нижньої оцінки, отриманої за допомогою напіввизначеної релаксації, для знаходження верхньої оцінки початкової задачі. Обґрунтовуються переваги напіввизначеної релаксації та досліджується її точність.

3.1 Розв'язування загальних квадратичних задач

3.1.1 Постановка задачі і метод розв'язування

Загальна задача квадратичної оптимізації має наступний вигляд:

$$\min \left\{ x^T Q x + d^T x \mid x^T A_i x + b_i^T x \leq c_i, i = 1, \dots, m \right\}, \quad (3.1)$$

де Q – симетрична матриця ($n \times n$);

A_i – симетричні матриці ($n \times n$);

x – вектор-стовпчик змінних розмірності n ;

b_i та d – вектор-стовпчики розмірності n ;

c – вектор-стовпчик розмірності m , де c_i – його компоненти.

Використаємо напіввизначену релаксацію [64] для розв'язування квадратичної задачі (3.1) (змістовні перетворення приведені у п.1.4.2). Отримаємо таку задачу напіввизначеної оптимізації для задачі квадратичного програмування (3.1):

$$\min \{ \tilde{Q} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, i=1, \dots, m, Y \succeq 0 \}, \quad (3.2)$$

$$\text{де } Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}; \quad (3.3)$$

$$\tilde{Q} = \begin{pmatrix} 0 & \frac{d^T}{2} \\ \frac{d}{2} & Q \end{pmatrix};$$

$$\tilde{A}_i = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix}.$$

Опустимо умову додатної напіввизначеності матриці Y у задачі (3.2) та будемо її розв'язувати напіввизначеним симплекс-методом (додатна напіввизначеність матриці Y визначається симплекс-методом). Якщо отримана матриця Y буде мати ранг одиниця, то отриманий розв'язок задачі (3.2) є глобальним мінімумом задачі (3.1). Якщо ж ранг матриці Y буде більшим одиниці, то ми отримали нижню оцінку цільової функції задачі (3.1). Тоді продовжимо розв'язувати початкову задачу (3.1) методом внутрішньої точки для локальної оптимізації, використовуючи у якості початкової точки отриманий розв'язок задачі (3.2). Таким чином, отримаємо верхню оцінку задачі (3.1). Чисельні експерименти показують, що верхня оцінка часто співпадає з розв'язком задачі (3.1). Таким чином, напіввизначена релаксація дозволяє отримувати розв'язки в околі точки глобального мінімуму задачі (3.1). Зауважимо, що така схема розв'язку загальних квадратичних задач раніше не досліджувалася.

3.1.2 Порівняльні чисельні експерименти

Розглянемо наступну квадратичну задачу:

$$\min\{\|x\|^2 - 8x_1^2 + 4x_2^2 - 6x_1x_2 + 4x_1 - 2x_2 - 14 \leq 0, -2x_1^2 - 8x_2^2 - 2x_1x_2 - x_1 - x_2 + 42 \leq 0\}.$$

На рис. 3.1 показана допустима множина цієї задачі. Ця множина є незв'язною та містить 4 локальних мінімумів. Перше обмеження ($-8x_1^2 + 4x_2^2 - 6x_1x_2 + 4x_1 - 2x_2 - 14 \leq 0$) є неопуклим (гіпербола), друге обмеження ($-2x_1^2 - 8x_2^2 - 2x_1x_2 - x_1 - x_2 + 42 \leq 0$) також є неопуклим (зовнішня частина еліпсу). Після перетворення цієї задачі до напіввизначеної за допомогою напіввизначеної релаксації та розв'язування її напіввизначеним симплекс-методом була одержана наступна матриця

$$\begin{pmatrix} 1 & 0,41399 & 2,15917 \\ 0,41399 & 0,17139 & 0,893883 \\ 2,15917 & 0,89388 & 4,662041 \end{pmatrix}$$

ранг якої дорівнює одиниці. Тому можна зробити висновок, що напіввизначена релаксація є точною, і була знайдена точка глобального мінімуму даної квадратичної задачі $x^* = (0,41399; 2,15917)$ з точністю до заданного ε .

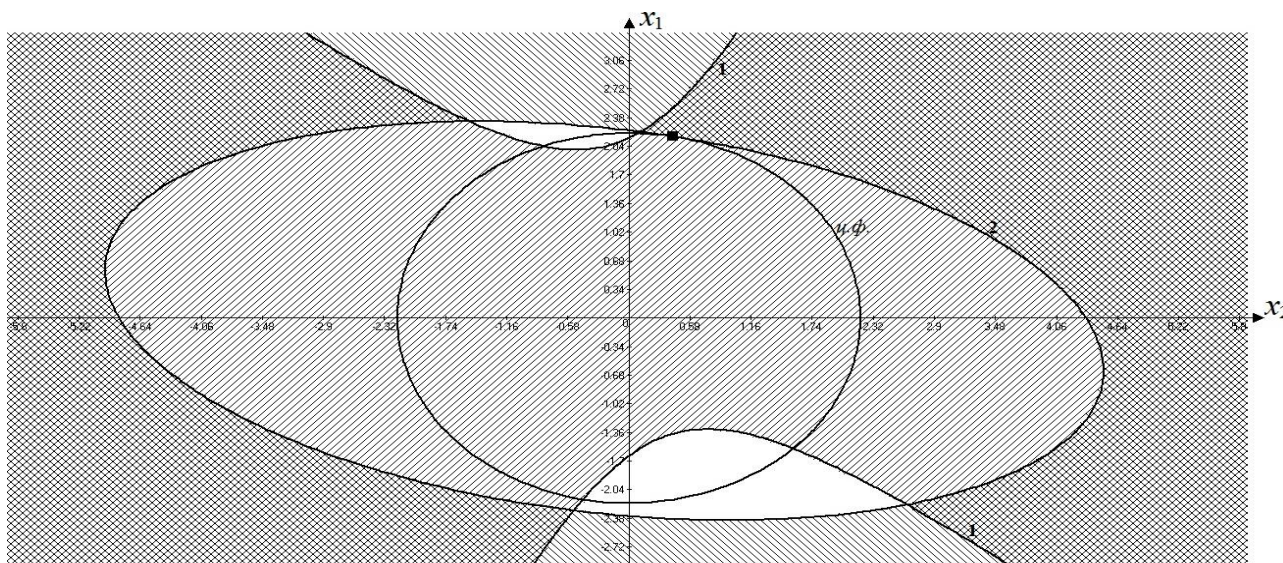


Рисунок 3.1 – Задача має 4 локальних мінімумів, де

 – допустима множина для обмеження $-8x_1^2 + 4x_2^2 - 6x_1x_2 + 4x_1 - 2x_2 - 14 \leq 0$;

 – допустима множина для обмеження $-2x_1^2 - 8x_2^2 - 2x_1x_2 - x_1 - x_2 + 42 \leq 0$;

 – допустима множина задачі;

1 – перше обмеження ($-8x_1^2 + 4x_2^2 - 6x_1x_2 + 4x_1 - 2x_2 - 14 = 0$);

2 – друге обмеження ($-2x_1^2 - 8x_2^2 - 2x_1x_2 - x_1 - x_2 + 42 = 0$);

ц.ф. – цільова функція;

▪ – точка глобального мінімуму задачі.

Розглянемо приклад загальної квадратичної задачі (3.1) з трьома обмеженнями. Представимо її у вигляді (3.2) (дані представлені в табл. Б.1.)

Результати розв'язування представлені в табл. 3.1.

Таблиця 3.1 – Матриця розв'язку

Y =	1	0,508	0,3544	0,151	1E-11	7E-08	0,159
	0,5079	0,372	0,2591	0,085	-0,08	-0,026	0,0872
	0,3544	0,259	0,1806	0,059	-0,06	-0,018	0,0608
	0,1515	0,085	0,0595	0,024	-0,01	-0,002	0,0246
	1E-11	-0,08	-0,056	-0,01	0,056	0,018	-0,004
	7E-08	-0,026	-0,018	0	0,018	0,006	-0,001
	0,159	0,087	0,0608	0,025	0	-0,001	0,0257

Ранг матриці розв'язку (табл. 3.1) дорівнює 2, тому можна зробити висновок, що нами була одержана нижня оцінка цільової функції задачі: 0,664244, яка не значно відрізняється від глобального мінімуму: 0,6724962.

У табл. 3.2 наведені результати чисельних експериментів по розв'язуванню загальних квадратичних задач різної розмірності та знаходженню нижньої оцінки їх цільових функцій (f_{SDP}).

Таблиця 3.2 – Результати чисельних експериментів

№п\п	n	ранг	f_{SDP}	f_{min}
1	6	2	0,53	0,67017
2	6	3	0,769978	1,470926
3	10	2	0,453065	0,45308
4	10	3	0,375502	0,575855
5	10	3	9,883102	12,34698
6	10	3	0,375502	0,575855
7	12	2	1,38959	1,539229
8	15	2	0,6795	0,874392

Як бачимо, ранг матриці розв'язку задач незначно відрізняється від одиниці. Ці експерименти показують, що напіввизначена релаксація може бути використана для пошуку нижньої оцінки цільової функції в загальних квадратичних задачах.

Розглянемо задачу s_2 розмірності 3×8 з [30]:

$$\begin{aligned}
 \min \quad & x_1, \\
 \text{s.t.} \quad & -x_1 - 2x_2x_3 \leq 0, \\
 & 2x_2 + 2x_3 + 4x_2x_3 \leq 3, \\
 & -3 \leq x_1 \leq 10, \\
 & 0 \leq x_2, x_3 \leq 1.
 \end{aligned} \tag{3.4}$$

Зменшимо кількість обмежень у задачі (3.4). Використовуючи властивість

$$\begin{aligned}
 \min \quad & x_1, \\
 \text{s.t.} \quad & f(x) \leq x_1,
 \end{aligned} \Leftrightarrow \min f(x),$$

перетворимо задачу (3.4) до вигляду

$$\begin{aligned}
\min \quad & -2x_2x_3, \\
\text{s.t.} \quad & 2x_2 + 2x_3 + 4x_2x_3 \leq 3, \\
& -3 \leq x_1 \leq 10, \\
& 0 \leq x_2, x_3 \leq 1.
\end{aligned} \tag{3.5}$$

Двосторонні обмеження на змінну вигляду $a \leq x \leq b$ перепишемо у вигляді $(x-a)(x-b) \leq 0$, та, розкривши дужки, отримаємо наступну квадратичну нерівність

$$x^2 + x(-a-b) \leq -ab. \tag{3.6}$$

Застосуємо (3.6) до (3.5). Отримаємо задачу

$$\begin{aligned}
\min \quad & -2x_2x_3, \\
\text{s.t.} \quad & 2x_2 + 2x_3 + 4x_2x_3 \leq 3, \\
& x_1^2 - 7x_1 \leq 30, \\
& x_2^2 - x_2 \leq 0, \\
& x_3^2 - x_3 \leq 0.
\end{aligned} \tag{3.7}$$

Таким чином, отримали загальну квадратичну задачу розмірності 3×4 , яка є еквівалентною до (3.4). Перепишемо (3.7) в загальному у вигляді:

$$\min \{ x^T Q x \mid x^T A_i x + b_i^T x \leq c_i, \quad i = 1, \dots, 4 \}, \tag{3.8}$$

$$\text{де } Q = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad c = \begin{pmatrix} 3 \\ 30 \\ 0 \\ 0 \end{pmatrix};$$

$$A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}, b_1 = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, b_2 = \begin{pmatrix} -7 \\ 0 \\ 0 \end{pmatrix};$$

$$A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, b_3 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, b_4 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Застосуємо напіввизначену релаксацію (3.2)-(3.3), описану вище, до задачі (3.8). Отримаємо

$$\min \{ \tilde{Q} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, i=1, \dots, 4, Y \succeq 0 \}, \quad (3.9)$$

$$\text{де } x^T Q x = \begin{pmatrix} 0 & 0 \\ 0 & Q \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & Q \end{pmatrix} \bullet Y = \tilde{Q} \bullet Y;$$

$$x^T A_i x + b_i^T x - c_i = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix} = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet Y = \tilde{A}_i \bullet Y;$$

$$Y = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}^T = \begin{pmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & x_1 x_1 & x_1 x_2 & x_1 x_3 \\ x_2 & x_2 x_1 & x_2 x_2 & x_2 x_3 \\ x_3 & x_3 x_1 & x_3 x_2 & x_3 x_3 \end{pmatrix};$$

$$\tilde{Q} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \tilde{A}_1 = \begin{pmatrix} -3 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}, \tilde{A}_2 = \begin{pmatrix} -30 & -3.5 & 0 & 0 \\ -3.5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{A}_3 = \begin{pmatrix} 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{A}_4 = \begin{pmatrix} 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 1 \end{pmatrix}.$$

Розв'язуємо задачу (3.9) напіввизначеним симплекс-методом. Зауважимо, що при розв'язуванні задачі (3.9) напіввизначеним симплекс-методом потрібно додати ще одне обмеження виду $\tilde{A}_5 \bullet Y = 1$, де

$$\tilde{A}_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

яке ми отримуємо з умови, що елемент $y_{11} = 1$.

Розв'язавши (3.9) напіввизначеним симплекс-методом, отримали значення цільової функції $-0,75$ та матрицю Y рангу два

$$Y = \begin{pmatrix} 1 & 0 & 0,375 & 0,375 \\ 0 & 0 & 0 & 0 \\ 0,375 & 0 & 0,375 & 0,375 \\ 0,375 & 0 & 0,375 & 0,375 \end{pmatrix}, \quad (3.10)$$

з чого можна зробити висновок, що ми отримали нижню оцінку цільової функції задачі (3.4). З (3.10) маємо шуканий вектор

$$x = \begin{pmatrix} 0 \\ 0,375 \\ 0,375 \end{pmatrix}.$$

Для отримання верхньої оцінки використаємо знайдену точку x в якості початкової та будемо розв'язувати задачу (3.4) методом внутрішньої точки. Отримаємо точку

$$x = \begin{pmatrix} -0,5 \\ 0,5 \\ 0,5 \end{pmatrix}$$

та значення мінімуму $-0,5$, яке є оптимальним.

Є ще один спосіб знаходження шуканої точки з матриці Y – це взяти квадратний корінь з елементів, які знаходяться на головній діагоналі Y , при чому знак відповідного елемента повинен співпадати зі знаком відповідного елемента вектора x , який ми знайшли вище. Для матриці (3.10) шуканий вектор, знайдений цим способом, буде мати вигляд

$$x = \begin{pmatrix} 0 \\ 0,612 \\ 0,612 \end{pmatrix}.$$

Розглянемо деякі тестові задачі, наведені в [30] і [25]. Для їх розв'язування використовувалася схема (3.2)-(3.3), описана вище.

Для розв'язування були обрані загальні квадратичні задачі, для яких знаходилася нижня оцінка напіввизначеним симплекс-методом, а потім проводився пошук верхньої оцінки методами локального пошуку: прямо-двоїстим методом внутрішньої точки. Результати чисельних експериментів наведені в табл.3.3 і табл.3.4. Як можна побачити, в 91,66% задач отримана верхня оцінка була глобальним мінімумом початкової задачі з заданою точністю. У задачах *fp_2_1* з [30] та *g15* з [25] було знайдено тільки наближені розв'язки. У деяких задачах знайдена нижня оцінка була оптимальною.

Таблиця 3.3 – Результати чисельних експериментів над задачами з [30]

Назва задачі в [30]	Розмірність перетвореної задачі	Отримана нижня оцінка	Отримана верхня оцінка	Оптимальний розв'язок з [30]
fp_2_1	6×7	-18,86	-16,5	-17
fp_2_2	7×9	-213	-213	-213
fp_2_4	7×12	-23.71	-11	-11
fp_3_3	7×13	-438	-310	-310
fp_3_4	3×6	-5	-4	-4
e_1	3×4	-3	-3	-3
f_a	3×5	-5,98	-1,083	-1,083
f_b	2×3	-8,572	-8,5	-8,5
f_c	5×11	-13	-13	-13
f_f	2×6	-2,828	-2,828	-2,828
s_1	3×5	0	0,74	0,74
s_1b	3×5	0	0,74	0,74
s_1c	3×5	0,69	0,74	0,74
s_1d	3×5	0,4	0,74	0,74
s_2	3×4	-0,75	-0,5	-0,5
s_2b	3×4	-1,5	-0,5	-0,5
s_2c	3×4	-0,54	-0,5	-0,5
s_2d	3×4	-0,938	-0,5	-0,5

Таблиця 3.4 – Результати чисельних експериментів над задачами з [25]

Назва задачі в [25]	Розмірність перетвореної задачі	Отримана нижня оцінка	Отримана верхня оцінка	Оптимальний розв'язок з [25]
g01	13×22	-15	-15	-15
g04	5×11	-32232	-30665	-30665
g07	10×18	24,3064	24,3062	24,3062

Продовження таблиці 3.4

Назва задачі в [25]	Розмірність перетвореної задачі	Отримана нижня оцінка	Отримана верхня оцінка	Оптимальний розв'язок з [25]
g11	3×4	0,75	0,75	0,75
g15	3×2	943,985	-	961,715
g18	10×23	-0,866	-0,866	-0,866

Таким чином, як показали чисельні експерименти, послідовне розв'язування релаксованої задачі напіввизначеним симплекс-методом та використання знайденої оптимальної точки в якості початкової для загальної квадратичної задачі, яку розв'язуємо методом внутрішньої точки, часто дозволяє знаходити точку глобального мінімуму загальної квадратичної задачі.

3.2 Уточнення напіввизначеної релаксації

Напіввизначену релаксацію будемо називати точною, якщо розв'язок відповідної задачі SDP однозначно визначає розв'язок початкової квадратичної задачі.

3.2.1 Розв'язування задачі максимізації норми вектора на перетині еліпсоїдів

Розглянемо загальну задачу квадратичної оптимізації:

$$\min \left\{ x^T Q x + p^T x \mid x^T A_i x + b_i^T x \leq c_i, i = 1, \dots, m \right\}, \quad (3.11)$$

де Q – симетрична матриця ($n \times n$);

A_i – симетричні матриці ($n \times n$);

x – вектор-стовпчик змінних розмірності n ;

b_i та p – вектор-стовпчики розмірності n ;

c – вектор-стовпчик розмірності m з компонентами c_i .

Використаємо метод точної квадратичної регуляризації [124] для розв'язування квадратичної задачі (3.11). Позначимо цільову функцію задачі (3.11) через $f_0(x)$, а функції обмежень через $f_i(x)$. Введемо нову змінну x_{n+1} . Розв'язок квадратичної релаксації задачі (3.11) буде мати вигляд

$$\tilde{x} = \begin{pmatrix} x_1 \\ \dots \\ x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} x \\ x_{n+1} \end{pmatrix},$$

де

$$x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$$

є розв'язок задачі (3.11).

Зведемо задачу (3.11) до вигляду

$$\min\{x_{n+1} \mid f_0(x) + s \leq x_{n+1}, f_i(x) \leq 0, i = 1, \dots, m\}, \quad (3.12)$$

де значення параметру s вибираємо таким, щоб

$$f_0(x^*) + s \geq \|x^*\|^2, \quad (3.13)$$

x^* – розв'язок задачі (3.11).

Далі, перетворенням простору $x = Az$, де матриця A порядку $(n+1) \times (n+1)$ має вигляд

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ z_1 & z_2 & \dots & z_{n+1} \end{pmatrix},$$

задачу (3.12) зведемо до наступної задачі

$$\min\{\|z\|^2 \mid f_0(\bar{z}) + s \leq \|z\|^2, f_i(\bar{z}) \leq 0, i = 1, \dots, m, z \in E^{n+1}\},$$

де $\bar{z} = (z_1, \dots, z_n)$;

$z = (\bar{z}, z_{n+1})$.

Існує таке значення параметра $r > 0$, що всі функції

$$g_0(z) = f_0(\bar{z}) + s + (r - 1)\|z\|^2, g_i(z) = f_i(\bar{z}) + r\|z\|^2, i = 1, \dots, m,$$

будуть опуклими [124].

Таким чином, задача (3.14) зведена до наступної

$$\min\{\|z\|^2 \mid g_i(z) \leq d, i = 0, \dots, m, r\|z\|^2 = d\}, \quad (3.14)$$

де всі $g_i(z)$ – опуклі функції; d – нова змінна. Доведено [124], що задача (3.14) еквівалентна опуклій задачі

$$\min\{d \mid g_i(z) \leq d, i = 0, \dots, m, r\|z\|^2 \leq d\},$$

або задачі

$$\max\{\|z\|^2 \mid g_i(z) \leq d, i = 0, \dots, m\}, \quad (3.15)$$

для мінімального значення змінної d , яке знаходимо методом дихотомії. Нехай z^* – розв’язок задачі (3.15) для мінімального допустимого значення d , отриманий прямо-двоїтим методом внутрішньої точки (розв’язок задачі (3.15) співпадає з розв’язком задачі (3.11), якщо виконується умова (3.13) [124]).

Визначаємо параметр s за формулою:

$$s \geq \|x^*\|^2 - f(x^*),$$

де $z^* = (x^*, x_{n+1}^*)$ – розв’язок задачі (3.15).

Далі знаходимо $x_{n+1} = \sqrt{f(x^*) + s - \|x^*\|^2}$ та $d = r \|z^*\|^2$.

Таким чином, отримали наступну задачу ($\tilde{x} = (x, x_{n+1})$)

$$\max \left\{ \begin{array}{l} \|\tilde{x}\|^2 \mid \tilde{x}^T Q \tilde{x} + p^T \tilde{x} + s + (r-1)(\|\tilde{x}\|^2) \leq d \\ \tilde{x}^T A_i \tilde{x} + b_i^T \tilde{x} - c_i + r \|\tilde{x}\|^2 \leq d, \quad i = 1, \dots, m \end{array} \right\},$$

або

$$\max \left\{ \begin{array}{l} \|\tilde{x}\|^2 \mid \tilde{x}^T Q \tilde{x} + p^T \tilde{x} + s + (r-1)(\|\tilde{x}\|^2) - d \leq 0 \\ \tilde{x}^T A_i \tilde{x} + b_i^T \tilde{x} - c_i + r \|\tilde{x}\|^2 - d \leq 0, \quad i = 1, \dots, m \end{array} \right\},$$

або

$$\min \left\{ \begin{array}{l} -\|\tilde{x}\|^2 \mid \tilde{x}^T Q \tilde{x} + p^T \tilde{x} + s + (r-1)(\|\tilde{x}\|^2) - d \leq 0 \\ \tilde{x}^T A_i \tilde{x} + b_i^T \tilde{x} - c_i + r \|\tilde{x}\|^2 - d \leq 0, \quad i = 1, \dots, m \end{array} \right\}, \quad (3.16)$$

яка є квадратичною. Застосуємо до неї напіввизначену релаксацію [64].

Введемо матрицю змінних Y , яка може бути виражена через будь-який вектор x таким чином [43]:

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}.$$

Таким чином, Y є симетричною та додатно напіввизначеною матрицею [43]. Використовуючи матрицю Y , перепишемо цільову функцію та функції обмежень задачі (3.16):

$$\|\tilde{x}\|^2 = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet Y, \quad (3.17)$$

де I – одинична матриця, а обмеження задачі (3.16) приймуть вигляд

$$\begin{aligned} & \tilde{x}^T Q \tilde{x} + p^T \tilde{x} + s + (r-1)(\|\tilde{x}\|^2) - d = \\ & = \begin{pmatrix} (s-d) & \frac{p^T}{2} & 0 \\ \frac{p}{2} & Q & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} + (r-1) \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} = \\ & = \begin{pmatrix} (s-d) & \frac{p^T}{2} & 0 \\ \frac{p}{2} & Q & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet Y + (r-1) \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet Y = \begin{pmatrix} (s-d) & \frac{p^T}{2} & 0 \\ \frac{p}{2} & Q & 0 \\ 0 & 0 & 0 \end{pmatrix} + (r-1) \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet Y, \quad (3.18) \end{aligned}$$

у разі, якщо матриця A_i не є додатно напіввизначеною (тобто відповідне

обмеження є неопуклим):

$$\begin{aligned}
 & \tilde{x}^T A_i \tilde{x} + b_i^T \tilde{x} - c_i + r \|\tilde{x}\|^2 - d = \\
 & = \begin{pmatrix} (-c_i - d) & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} + r \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} = \\
 & = \begin{pmatrix} (-c_i - d) & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet Y + r \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet Y = \begin{pmatrix} (-c_i - d) & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} + r \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \bullet Y, \quad (3.19)
 \end{aligned}$$

або, у разі, якщо матриця A_i є додатно напіввизначеною (тобто відповідне обмеження є опуклим):

$$\begin{aligned}
 & \tilde{x}^T A_i \tilde{x} + b_i^T \tilde{x} - c_i + r \|\tilde{x}\|^2 - d = \\
 & = \begin{pmatrix} -c_i & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{x}\tilde{x}^T \end{pmatrix} = \begin{pmatrix} -c_i & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet Y. \quad (3.20)
 \end{aligned}$$

Введемо позначення

$$\tilde{C} = \begin{pmatrix} 0 & 0 \\ 0 & -I \end{pmatrix}, \text{ де } I - \text{одинична матриця,}$$

$$\tilde{A}_i = \begin{pmatrix} \begin{pmatrix} (-c_i - d) & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} + r \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \\ \end{pmatrix}, \quad i = 1, \dots, m, \quad (3.21)$$

$$\tilde{A}_{m+1} = \begin{pmatrix} \begin{pmatrix} (s - d) & \frac{p^T}{2} \\ \frac{p}{2} & Q \end{pmatrix} + (r - 1) \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \\ \end{pmatrix}.$$

Тоді маємо таку задачу напіввизначеної оптимізації для задачі квадратичного програмування (3.12):

$$\min \left\{ \tilde{C} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, \quad i = 1, \dots, m+1, \quad Y \succeq 0 \right\}, \quad (3.22)$$

яку розв'яжемо напіввизначеним симплекс-методом та отримаємо нижню оцінку цільової функції задачі (3.11).

3.2.2 Порівняльні чисельні експерименти

Розглянемо задачу s_2 розмірності 3×8 з [30]:

$$\begin{aligned} \min \quad & x_1, \\ \text{s.t.} \quad & -x_1 - 2x_2x_3 \leq 0, \\ & 2x_2 + 2x_3 + 4x_2x_3 \leq 3, \\ & -3 \leq x_1 \leq 10, \\ & 0 \leq x_2, x_3 \leq 1. \end{aligned} \quad (3.23)$$

Перепишемо (3.23) в загальному вигляді (див. п. 3.1.2):

$$\min \{ x^T Q x \mid x^T A_i x + b_i^T x \leq c_i, \quad i = 1, \dots, 4 \}, \quad (3.24)$$

$$\text{де } Q = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad c = \begin{pmatrix} 3 \\ 30 \\ 0 \\ 0 \end{pmatrix};$$

$$A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}, \quad b_1 = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad b_2 = \begin{pmatrix} -7 \\ 0 \\ 0 \end{pmatrix};$$

$$A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad b_3 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad b_4 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Використаємо для задачі (3.24) напіввизначену релаксацію (3.9), описану вище:

$$\min \{ \tilde{Q} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, \quad i = 1, \dots, 4, \quad Y \succeq 0 \}, \quad (3.25)$$

$$\text{де } x^T Q x = \begin{pmatrix} 0 & 0 \\ 0 & Q \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & Q \end{pmatrix} \bullet Y = \tilde{Q} \bullet Y;$$

$$x^T A_i x + b_i^T x - c_i = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix} = \begin{pmatrix} -c_i & \frac{b_i^T}{2} \\ \frac{b_i}{2} & A_i \end{pmatrix} \bullet Y = \tilde{A}_i \bullet Y;$$

$$Y = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}^T = \begin{pmatrix} 1 & x_1 & x_2 & x_3 \\ x_1 & x_1x_1 & x_1x_2 & x_1x_3 \\ x_2 & x_2x_1 & x_2x_2 & x_2x_3 \\ x_3 & x_3x_1 & x_3x_2 & x_3x_3 \end{pmatrix};$$

$$\tilde{Q} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \tilde{A}_1 = \begin{pmatrix} -3 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}, \tilde{A}_2 = \begin{pmatrix} -30 & -3.5 & 0 & 0 \\ -3.5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{A}_3 = \begin{pmatrix} 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tilde{A}_4 = \begin{pmatrix} 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 1 \end{pmatrix}.$$

Перетворимо задачу (3.25) до задачі максимізації норми вектора на перетині еліпсоїдів. Використаємо описану у п.3.2.1 схему (3.16–3.22).

Розрахуємо параметри s , r та d .

$$s = \|x^*\|^2 - f(x^*) + 1 = (-0,5)^2 + 0,5^2 + 0,5^2 - (-0,5) + 1 = 1,5,$$

$$r = 32, d = r \|x^*\|^2 = 32((-0,5)^2 + 0,5^2 + 0,5^2) = 24.$$

Отримаємо задачу напіввизначеної оптимізації

$$\min \{ \tilde{C} \bullet Y \mid \tilde{A}_i \bullet Y \leq 0, i=1, \dots, 5, Y \succeq 0 \}, \quad (3.26)$$

$$\text{де } Y = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}^T = \begin{pmatrix} 1 & x_1 & x_2 & x_3 & x_4 \\ x_1 & x_1x_1 & x_1x_2 & x_1x_3 & x_1x_4 \\ x_2 & x_2x_1 & x_2x_2 & x_2x_3 & x_2x_4 \\ x_3 & x_3x_1 & x_3x_2 & x_3x_3 & x_3x_4 \\ x_4 & x_4x_1 & x_4x_2 & x_4x_3 & x_4x_4 \end{pmatrix};$$

$$\tilde{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix};$$

$$\begin{aligned} \tilde{A}_1 &= \begin{pmatrix} (-c_i-d) & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} + r \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} -27 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 32 & 0 & 0 & 0 \\ 0 & 0 & 32 & 0 & 0 \\ 0 & 0 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 & 32 \end{pmatrix} = \\ &= \begin{pmatrix} -27 & 0 & 1 & 1 & 0 \\ 0 & 32 & 0 & 0 & 0 \\ 1 & 0 & 32 & 2 & 0 \\ 1 & 0 & 2 & 32 & 0 \\ 0 & 0 & 0 & 0 & 32 \end{pmatrix}; \end{aligned}$$

$$\tilde{A}_2 = \begin{pmatrix} -c_i & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -30 & -3.5 & 0 & 0 & 0 \\ -3.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{A}_3 = \begin{pmatrix} -c_i & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{A}_4 = \begin{pmatrix} -c_i & \frac{b_i^T}{2} & 0 \\ \frac{b_i}{2} & A_i & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{A}_5 = \begin{pmatrix} (s-d) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & 0 \end{pmatrix} + (r-1) \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} -22,5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 \\ 0 & 0 & 31 & 0 & 0 \\ 0 & 0 & 0 & 31 & 0 \\ 0 & 0 & 0 & 0 & 31 \end{pmatrix} =$$

$$= \begin{pmatrix} -22,5 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 \\ 0 & 0 & 31 & -1 & 0 \\ 0 & 0 & -1 & 31 & 0 \\ 0 & 0 & 0 & 0 & 31 \end{pmatrix}.$$

Зазначимо, що для розрахунку \tilde{A}_1 використовувалась формула (3.19), так як це обмеження є неопуклим (матриця A_1 з (3.24) не є додатно напіввизначеною). Для розрахунку \tilde{A}_2 , \tilde{A}_3 та \tilde{A}_4 використовувалась формула (3.20), так як ці обмеження є опуклими (матриці A_2 , A_3 та A_4 з (3.38) є додатно напіввизначеними). Для розрахунку \tilde{A}_5 використовувалась формула (3.18).

Кількість обмежень задачі (3.26) у порівнянні з (3.25) збільшилась на 1, та розмірність задачі (3.26) у порівнянні з (3.25) збільшилась також на 1.

Розв'язуємо задачу (3.26) напіввизначеним симплекс-методом. Зауважимо, що при розв'язуванні задачі (3.26) напіввизначеним симплекс-методом потрібно додати ще одне обмеження виду $\tilde{A}_6 \bullet Y = 1$, де

$$\tilde{A}_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

яке ми отримуємо з умови, що елемент $y_{11} = 1$ у матриці Y .

Розв'язавши (3.26) напіввизначеним симплекс-методом, отримали наступну матрицю Y

$$Y = \begin{pmatrix} 1,0000 & 0,0677 & 0,3670 & 0,3680 & 0,0677 \\ 0,0677 & 0,0074 & 0,0400 & 0,0401 & 0,0074 \\ 0,3670 & 0,0400 & 0,3668 & 0,3674 & 0,0400 \\ 0,3680 & 0,0401 & 0,3674 & 0,3680 & 0,0401 \\ 0,0677 & 0,0074 & 0,0400 & 0,0401 & 0,0074 \end{pmatrix}. \quad (3.27)$$

Відкидаємо останній рядок та останній стовпчик у (3.27), отримаємо матрицю розмірністю 4×4

$$Y = \begin{pmatrix} 1,0000 & 0,0677 & 0,3670 & 0,3680 \\ 0,0677 & 0,0074 & 0,0400 & 0,0401 \\ 0,3670 & 0,0400 & 0,3668 & 0,3674 \\ 0,3680 & 0,0401 & 0,3674 & 0,3680 \end{pmatrix}, \quad (3.28)$$

яку підставляємо у цільову функцію задачі (3.25) та отримуємо нижню оцінку цільової функції задачі (3.23), яка дорівнює $-0,7347$ та є кращою від нижньої оцінки, отриманої методом напіввизначеної релаксації для цієї задачі у п.3.1.2.

З (3.28) маємо шуканий вектор

$$x = \begin{pmatrix} 0,0677 \\ 0,3670 \\ 0,3680 \end{pmatrix}, \quad (3.29)$$

або візьмемо квадратний корінь з елементів, які знаходяться на головній діагоналі Y (3.28), при чому знак відповідного елемента повинен співпадати зі знаком відповідного елемента вектора x , який ми знайшли вище, і отримаємо точку

$$x = \begin{pmatrix} 0,0860 \\ 0,6056 \\ 0,6056 \end{pmatrix}.$$

Для отримання верхньої оцінки задачі (3.23) використаємо знайдену точку x (3.29) в якості початкової та будемо розв'язувати задачу (3.23) методом внутрішньої точки. Отримаємо точку

$$x = \begin{pmatrix} -0,5 \\ 0,5 \\ 0,5 \end{pmatrix}$$

та значення мінімуму $-0,5$, яке є глобальним.

Порівняємо значення нижніх оцінок, отриманих за допомогою напіввизначеної релаксації, з нижніми оцінками, отриманими методом максимізації норми вектора на перетині еліпсоїдів в деяких задачах з [30]. Як можна побачити з табл. 3.5, метод максимізації норми вектора на перетині еліпсоїдів дає кращу нижню оцінку, ніж метод напіввизначеної релаксації.

Таблиця 3.5 – Результати чисельних експериментів над задачами з [30]

Назва задачі в [30]	Отримана нижня оцінка методом максимізації норми вектора на перетині еліпсоїдів	Отримана нижня оцінка за допомогою напіввизначеної релаксації
fp_3_4	-5,08	-5
f_b	-8,537	-8,572
s_2	-0,735	-0,75
s_2d	-0,828	-0,938

3.2.3 Задачі, для яких напіввизначена релаксація є точною

Нехай P – прямокутний паралелепіпед, який можна представити у вигляді

$$P = \{x | (x_i - a_i)(x_i - b_i) \leq 0, i = 1, \dots, n\}.$$

Тоді буде справедливим наступне твердження.

Теорема 3.1. *Напіввизначена релаксація для квадратичної задачі*

$$\max\{\|x\|^2 | x \in P\} \tag{3.30}$$

є точною.

Доведення. Припустимо протилежне, а саме що ранг матриці X (відповідної задачі SDP) більше одиниці та дорівнює k . Так як обмеження задачі (3.30) незалежні, то достатньо розглянути перше обмеження задачі SDP

$$\begin{pmatrix} 0 & -(a_1 + b_1)/2 \\ -(a_1 + b_1)/2 & 1 \end{pmatrix} \bullet \left[\begin{pmatrix} 1 & x_{11} \\ x_{11} & x_{11}^2 \end{pmatrix} + \dots + \begin{pmatrix} 1 & x_{1k} \\ x_{1k} & x_{1k}^2 \end{pmatrix} \right] = -a_1 b_1,$$

звідки

$$x_{11}^2 - (a_1 + b_1)x_{11} + \dots + x_{1k}^2 - (a_1 + b_1)x_{1k} = -a_1 b_1.$$

Отримано рівняння кулі, яка симетрична щодо системи координат. Її центр лежить на прямій $x = t$ (всі компоненти центру співпадають). Тоді точка цієї кулі, найбільш віддалена від початку координат, буде також лежати на прямій $x = t$. Це означає, що максимум норми вектора x досягається при $x_{11} = \dots = x_{1k}$. Таким чином, матриця в квадратних дужках матиме ранг одиниця. Отримане протиріччя доводить теорему.

Розглянемо іншу опуклу множину – симплекс, який задамо наступним чином

$$\Delta = \{x | x_i(x_i - a) \leq 0, i=1, \dots, n, c^T x = 1\}.$$

Для цієї опуклої множини справедливе аналогічне твердження.

Теорема 3.2. *Напіввизначена релаксація для квадратичної задачі*

$$\max\{\|x\|^2 | x \in \Delta\} \tag{3.31}$$

є точною.

Доведення. Розглянемо обмеження

$$\begin{pmatrix} 0 & \dots & 0 & -a/2 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ -a/2 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & x_1 & x_2 & \dots & x_n \\ x_1 & x_{11} & x_{12} & \dots & x_{1n} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \leq 0,$$

$$\begin{pmatrix} 0 & c_1/2 & c_2/2 & \dots & c_n/2 \\ c_1/2 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \bullet \begin{pmatrix} 1 & x_1 & x_2 & \dots & x_n \\ x_1 & x_{11} & x_{12} & \dots & x_{1n} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} = 1.$$

Ці обмеження рівносильні обмеженням

$$x_{ii} - ax_i \leq 0, i = 1, \dots, n,$$

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = 1$$

відповідно. Нехай розв'язок відповідної задачі SDP досягається у точці x , тоді ця точка є опуклою комбінацією крайніх точок симплексу

$$x = \sum_{i=1}^n \frac{\alpha_i}{c_i}, \quad \sum_{i=1}^n \alpha_i = 1, \alpha \geq 0.$$

Тоді задача SDP рівносильна наступній

$$\max \left\{ \sum_{i=1}^n \frac{\alpha_i}{c_i} \mid \sum_{i=1}^n \alpha_i = 1, \alpha \geq 0 \right\}.$$

Розв'язок цієї задачі дорівнює

$$x_k = \frac{1}{\min c_i},$$

а $x_i = 0, \forall i \neq k$ (що співпадає з розв'язком задачі (3.31)) але тоді

$$x_{kk} = a \frac{1}{\min c_i},$$

звідки випливає, що в загальному випадку, $x_k^2 \neq x_{kk}$. Це означає, що ранг матриці X не дорівнюватиме 1, але напіввизначена релаксація буде точною, так як розв'язок задачі (3.31) визначається першим рядком матриці X . Теорема доведена.

Зауваження 3.1. На відміну від паралелепіпеда, відповідна задача SDP

матиме ранг 2. Це означає, що вимога рівності ранга матриці задачі SDP одиниці, при якому напіввизначена релаксація буде точною, є тільки достатньою, але не необхідною умовою.

Покажемо, що обмеження на змінні (паралелепіпед) задачі можна використовувати для скорочення допустимої області задачі SDP та збільшення точності напіввизначеної релаксації.

Розглянемо приклад квадратичної задачі

$$\min\{\|x\|^2 \mid -4x_1^2 + 2x_2^2 + 5x_1 - 6x_2 + 10 \leq 0, -2x_1^2 - 4x_2^2 - 8x_1 + 6x_2 + 5 \leq 0\}.$$

На рис. 3.2 бачимо, що точка глобального мінімуму дорівнює $x^* = (-1.694, -1.194)$. Крім того, в цій задачі є ще 2 локальних мінімуми.

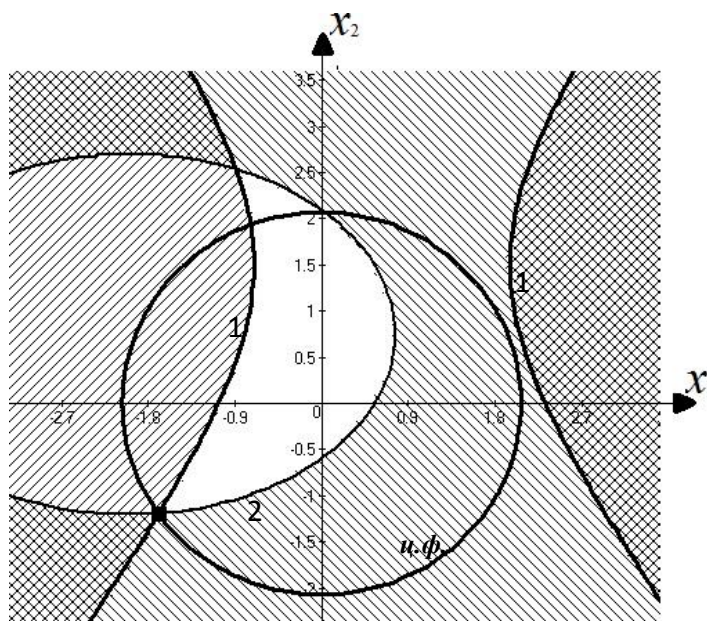


Рисунок 3.2 – Загальна квадратична задача, де

– допустима множина для обмеження $-4x_1^2 + 2x_2^2 + 5x_1 - 6x_2 + 10 \leq 0$;

– допустима множина для обмеження $-2x_1^2 - 4x_2^2 - 8x_1 + 6x_2 + 5 \leq 0$;

– допустима множина задачі;

1 – перше обмеження ($-4x_1^2 + 2x_2^2 + 5x_1 - 6x_2 + 10 = 0$);

2 – друге обмеження ($-2x_1^2 - 4x_2^2 - 8x_1 + 6x_2 + 5 = 0$);

ц.ф. – цільова функція;

▪ – точка глобального мінімуму задачі.

Перетворимо цю задачу до напіввизначеної та отримаємо розв'язок

$$\begin{pmatrix} 1 & 0,16963 & 0,22638 \\ 0,16963 & 2,398097 & 0,04015 \\ 0,226382 & 0,040149 & 0,05126 \end{pmatrix}$$

Ранг цієї матриці дорівнює 2, а знайдене значення цільової функції є нижньою оцінкою. Тепер додамо обмеження на змінні $(x_1 + 3)(x_1 - 0.5) \leq 0$, $(x_2 + 3)(x_2 - 0.5) \leq 0$, що задовольняють знайденому розв'язку. При цих обмеженнях отримуємо новий розв'язок

$$\begin{pmatrix} 1 & -0,3285 & 0,188406 \\ -0,3285 & 2,321256 & -0,29952 \\ 0,188406 & -0,29952 & 1,028986 \end{pmatrix}$$

який також є наближенням, але кращим за попередній. Якщо обмеження на змінні взяти у вигляді $(x_1 + 1.8)x_1 \leq 0$, $(x_2 + 1.2)x_2 \leq 0$, що також задовольняють знайденому розв'язку, то отримаємо розв'язок

$$\begin{pmatrix} 1 & -1,58879 & -1,12932 \\ -1,58879 & 2,85983 & 1,768948 \\ -1,12932 & 1,768948 & 1,303701 \end{pmatrix}$$

що майже співпадає з розв'язком задачі.

Розглянемо задачу

$$\max \{ \|x\|^2 \mid Ax = b, x \geq 0 \}. \quad (3.32)$$

Як і раніше, представимо її у вигляді

$$\max\{\|x\|^2 \mid Ax = b, x_i(x_i - c_i) \leq 0, i = 1, \dots, n\}, \quad (3.33)$$

де значення c_1, \dots, c_n обираємо таким чином, щоб розв'язок задачі (3.32) був допустимий для задачі (3.33). Тоді справедливо наступне твердження.

Теорема 3.3. Нехай x^* – розв'язок задачі (3.33) та нерівність $c^T x^* \geq c^T x$ виконується для усіх допустимих x , тоді напіввизначена релаксація для задачі (3.33) буде точною.

Доведення. Нерівність $c^T x^* \geq c^T x$ буде виконуватися тоді і тільки тоді, коли x^* – також розв'язок задачі

$$\max\{c^T x \mid Ax = b, x \geq 0\}.$$

Якщо нерівність $\|x^*\|^2 \geq \|x\|^2$ для усіх допустимих x виконується, то точка x^* буде також розв'язком задачі (3.32). Нерівності $x_i(x_i - c_i) \leq 0, i = 1, \dots, n$ для відповідної задачі SDP будуть мати вигляд $x_{ii} \leq c_i x_i, i = 1, \dots, n$, а цільова функція

$$\max \sum_{i=1}^n x_{ii} \leq \sum_{i=1}^n c_i x_i,$$

звідки

$$\max \sum_{i=1}^n x_{ii} = \sum_{i=1}^n c_i x_i^*.$$

Це означає, що розв'язок відповідної задачі SDP однозначно визначає розв'язок задачі (3.33). Теорема доведена.

3.3 Розв'язування квадратичних задач з булевими змінними

3.3.1 Постановка задачі і метод розв'язування

Розглянемо задачу булевої квадратичної оптимізації

$$\min \{ c^T x \mid Ax \leq b, x_i = 0 \vee 1, x \in R^n \}, \quad (3.34)$$

де c – n -вимірний вектор;

A – матриця розміру $(m \times n)$;

x – n -вимірний вектор з компонентами 0 або 1;

R^n – n -вимірний евклідовий простір.

Запропонована методика розв'язування задач застосовна і для більш загальної задачі квадратичної оптимізації з квадратичними обмеженнями та булевими змінними

$$\min \{ x^T Q_0 x + q_0^T x \mid x^T Q_i x + q_i^T x \leq r_i, i = 1, \dots, m, x_i = 0 \vee 1, x \in R^n \}, \quad (3.35)$$

де усі Q_i – симетричні матриці розміру $(n \times n)$;

q_i, x – n -вимірні вектора;

r_i – скаляр.

Замінімо в задачах (3.34-3.35) булеві змінні квадратичною умовою $x_i(x_i - 1) = 0, i = 1, \dots, n$, де тепер змінні x_i можуть приймати довільні значення. Звичайно, що допустимими будуть тільки значення 0 або 1.

Таким чином, задачі (3.34-3.35) перетворюються до загальної квадратичної задачі

$$\min \left\{ x^T Q_0 x + q_0^T x \mid \begin{array}{l} x^T Q_i x + q_i^T x \leq r_i, i = 1, \dots, m, \\ x^T Q_i x + q_i^T x = r_i, i = m + 1, \dots, p, x \in E^n \end{array} \right\}. \quad (3.36)$$

Використаємо для її розв'язування напіввизначену релаксацію [64]. Враховуючи, що $x^T Q x = Q \bullet x x^T = Q \bullet X$ [18], перетворимо задачу (3.36) до задачі напіввизначеної оптимізації

$$\min \left\{ \begin{array}{l} \bar{A}_0 \bullet X \mid \bar{A}_i \bullet X \leq 0, \quad i = 1, \dots, m, \\ \bar{A}_i \bullet X = 0, \quad i = m + 1, \dots, p, \\ X \succeq 0 \end{array} \right\}, \quad (3.37)$$

$$\text{де } X = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix};$$

$$x^T Q_i x + q_i^T x - r_i = \begin{pmatrix} -r_i & \frac{q_i^T}{2} \\ \frac{q_i}{2} & Q_i \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & x x^T \end{pmatrix} = \begin{pmatrix} -r_i & \frac{q_i^T}{2} \\ \frac{q_i}{2} & Q_i \end{pmatrix} \bullet X = \bar{A}_i \bullet X, \quad i = 0, 1, \dots, p.$$

Перетворена задача (3.37) буде еквівалентною до задачі (3.36), якщо розв'язком задачі (3.37) буде додатно напіввизначена матриця X^* рангу одиниця. Ввести в обмеження задачі (3.37) умову, що ранг матриці X повинен дорівнювати одиниці, неможливо, так як ця умова перевіряється алгоритмічно. Тому дану умову в задачі (3.37) опускають, залишаючи тільки додатну напіввизначеність матриці X . Звичайно, таке ослаблення обмежень може призводити до зменшення цільової функції і до неприпустимого розв'язку, якщо розв'язок задачі (3.37) досягається не в крайньому промені конуса додатно напіввизначених матриць.

Для знаходження верхньої оцінки запишемо задачу (3.34-3.35) в загальному вигляді

$$\min \{ f_0(x) \mid f_i(x) \leq 0, i = 1, \dots, p, x \in E^n \}, \quad (3.38)$$

де всі $f_i(x)$ – квадратичні функції (обмеження-рівності можна замінити двома обмеженнями нерівностями).

Знайдений розв'язок задачі (3.37) використовуємо в якості початкової точки для розв'язування задачі (3.34-3.35) прямо-двоїстим методом внутрішньої точки [84]. Перетворимо обмеження задачі (3.38) до рівностей

$$\min\{f_0(x) - \mu \sum_{i=1}^p \ln(z_i) \mid f_i(x) + z_i = 0, i = 1, \dots, p\},$$

тоді функція Лагранжа цієї задачі буде мати вигляд

$$L(x, y, z) = f_0(x) - \mu \sum_{i=1}^p \ln(z_i) - \sum_{i=1}^p y_i (f_i(x) + z_i),$$

умови мінімуму якої дорівнюють

$$\nabla f_0(x) - \nabla f_i(x)^T y = 0,$$

$$f_i(x) + z_i = 0, i = 1, \dots, p,$$

$$-\mu Z^{-1} e + y = 0.$$

Метод Ньютона для цієї нелінійної системи рівнянь на кожній ітерації розв'язує лінійну систему рівнянь

$$\begin{bmatrix} G(x, y) & A^T(x) & 0 \\ A(x) & 0 & Z \\ A(x) & -I & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f_0(x) - A^T(x)y \\ -f(x) - z \\ \mu e - ZYe \end{bmatrix}, \quad (3.39)$$

де $G(x, y)$ – гессіан функції Лагранжа;

$$A(x) = \nabla f(x);$$

$$Y = \text{diag}(y_1, \dots, y_p);$$

$$Z = \text{diag}(z_1, \dots, z_p);$$

$$f(x) = (f_1(x), \dots, f_p(x));$$

I – одинична матриця;

$$e = (1, \dots, 1).$$

Розв'язок лінійної системи (3.39) використовуємо для переходу в наступну точку

$$x^{k+1} = x^k + \alpha_k \Delta x^k,$$

$$y^{k+1} = y^k + \alpha_k \Delta y^k,$$

$$z^{k+1} = z^k + \alpha_k \Delta z^k.$$

На кожній ітерації значення параметра μ спадає за формулою $\mu = x^T z / n$. Параметр α обирається так, щоб $z^{k+1} \geq 0$. Показано, що даний прямо-двоїстий метод внутрішньої точки глобально сходиться до точки локального мінімуму за поліноміальний час [71].

3.3.2 Порівняльні чисельні експерименти

Розглянемо деякі тестові задачі, наведені у [68], для розв'язання задачі про рюкзак

$$\max\{p^T x \mid a^T x \leq b, x_i = 0 \vee 1\}. \quad (3.40)$$

Верхня оцінка розв'язку x^0 цієї задачі, отримана напіввизначеним симплекс-методом, уточнювалася за допомогою розв'язування наступної задачі

$$\min\{\|x - x^0\|^2 \mid a^T x = b, x_i(x_i - 1) = 0, i = 1, \dots, n\}. \quad (3.41)$$

прямо-двоїтим методом внутрішньої точки.

Результати чисельних експериментів наведені в табл.3.6, де можна побачити, що розв'язок задачі (3.41) збігається з оптимальним розв'язком задачі (3.40).

Таблиця 3.6 – Результати чисельних експериментів над задачами з [68]

Задача (3.40) в [68]	Розмірність задачі	Отримана верхня оцінка (задача (3.37))	Отримана нижня оцінка (задача (3.41))	Оптимальний розв'язок
2.1	8	294,988	280	280
2.2	7	107,5393	107	107
2.3	6	159,997	150	150
2.4	7	127,0274	127	127
2.5	8	1189,51	900	900

Тобто, як видно з чисельних експериментів, розв'язок, отриманий методом внутрішньої точки, співпадає з розв'язком початкової задачі лінійної оптимізації з булевими змінними.

3.4 Переваги напіввизначеної релаксації

Таким чином, можна виділити такі переваги напіввизначеної релаксації над іншими методами для розв'язування загальних задач квадратичної оптимізації:

- напіввизначена релаксація дозволяє знайти нижню оцінку розв'язку або глобальний мінімум в загальних задачах квадратичної оптимізації незалежно від опуклості/неопуклості цільової функції та обмежень. На відміну від напіввизначеної релаксації, наприклад, метод спряжених градієнтів може бути використаний тільки для задачі опуклого квадратичного програмування. Методи гілок та границь розв'язують загальні задачі, але потребують експоненційного часу для знаходження розв'язку;

– перевірка оптимальності знайденого розв’язку на відміну від інших методів не потребує експоненційного часу та є досить простою (достатньо перевірити ранг матриці розв’язку X);

– напіввизначена релаксація дозволяє отримувати розв’язки в околі точки глобального мінімуму початкової задачі. А, наприклад, методи випадкового пошуку інколи знаходять оптимальну точку, але в багатьох випадках ці розв’язки далекі від оптимальних.

3.5 Висновки розділу

Основними результатами розділу є:

– уперше використано точну квадратичну регуляризацію для перетворення загальних квадратичних задач до задач максимізації норми вектора на перетині еліпсоїдів. Зокрема, проведено порівняння нижніх оцінок, отриманих за допомогою напіввизначеної релаксації початкової задачі, з нижніми оцінками, отриманими методом максимізації норми вектора на перетині еліпсоїдів, та виявлено, що метод максимізації норми вектора на перетині еліпсоїдів дає змогу знаходити кращу нижню оцінку цільової функції початкової задачі, ніж метод напіввизначеної релаксації (це пов’язано з тим, що перетин еліпсоїдів є опуклою множиною);

– введено нове поняття точності напіввизначеної релаксації та для окремих класів задач максимізації норми вектора на опуклій множині доведена точність напіввизначеної релаксації;

– уперше застосовано напіввизначений симплекс-метод для розв’язування задач напіввизначеної оптимізації, які отримуються шляхом релаксації загальних квадратичних задач, що дало змогу знаходити кращу нижню оцінку розв’язку початкової задачі;

– уперше запропоновано та перевірено на практиці нову процедуру для знаходження верхніх і нижніх оцінок цільової функції у загальних задачах квадратичної оптимізації, яка ґрунтується на використанні нижньої оцінки

напіввизначеної релаксації у якості початкової точки при розв'язуванні початкової задачі методами локального пошуку. Проведені чисельні експерименти для відомих тестових задач показали, що в 91,66% випадків отримана таким чином верхня оцінка співпадала з глобальним мінімумом початкової задачі.

Основні результати розділу опубліковано в [51, 122, 128, 136, 137, 142, 154, 156].

РОЗДІЛ 4

ПРИКЛАДНІ ЗАДАЧІ НАПІВВИЗНАЧЕНОЇ ОПТИМІЗАЦІЇ

У цьому розділі описується та удосконалюється використання напіввизначеної релаксації для широкого кола прикладних задач напіввизначеного програмування: задачі кластеризації даних, задачі розміщення сенсорних датчиків у мережі та задачі пошуку максимального розрізу графа, наводяться чисельні експерименти для цих класів задач. Уперше прикладні задачі зводяться до задачі максимізації норми вектора на перетині еліпсоїдів.

4.1 Модель кластеризації даних

4.1.1 Постановка задачі і метод розв'язування

Однією з основних задач штучного інтелекту та теорії розпізнавання образів є розбиття даних на кластери [100, 164, 166]. Необхідно об'єднати дані в групи однотипних об'єктів – кластери. Як правило, дані (об'єкти) представляються точками в n -вимірному просторі, які необхідно розбити на задане число підмножин, покриваючи їх шарами, які не перетинаються. Компоненти точок визначають параметри, які характеризують об'єкти. Відстань між двома точками всередині одного кластера завжди менше відстані між точками різних кластерів. У якості відстані може вибиратися різноманітна метрика простору. Існують ефективні методи розбиття даних на два кластери. Однак розбивка множини точок на більше ніж два кластери являє собою складну задачу [100, 146, 164, 166]. Це пов'язано з тим, що оптимальне розбиття точок на кластери зводиться до розв'язування багатоекстремальної задачі, в якій необхідно знайти точку глобального екстремуму. В даний час для розв'язування таких задач частіше використовуються генетичні або еволюційні методи, які засновані на випадковому пошуку, які дозволяють знаходити оптимальний розв'язок задач кластеризації тільки з деякою вірогідністю [47, 162]. Крім того, ці методи містять велику кількість параметрів, від значень яких

залежить їх ефективність.

Розглянемо множину m точок $\{a^1, \dots, a^m\}$ в n -вимірному евклідовому просторі. Необхідно розбити цю множину на 2 кластери таким чином, щоб кожна точка потрапила тільки в один кластер.

Розглянемо неорієнтований граф G з вершинами $M = \{1, \dots, m\}$ та множиною дуг E . Точки $\{a^1, \dots, a^m\}$ є вершинами цього графа. Нехай $w_{ij} = w_{ji}$ – вага дуги (a_i, a_j) . Для нашого випадку вага дуги w_{ij} є відстанню між точками a_i та a_j . Для заданого цілого $k \in 1, \dots, m-1$ задача k -clustering полягає у знаходженні підмножини S , яка включає k вершин, таких, що сумарна вага дуг підграфу, породженого S , є максимальною. Для незваженого графу ця проблема називається проблемою щільності підграфу (the densest subgraph problem).

Нехай W – матриця ваг дуг графа G . Тоді задачу k -clustering можна записати у вигляді задачі булевої оптимізації

$$\max \left\{ \frac{1}{2} x^T W x \mid \sum_{i=1}^m x_i = k, x_i = 0 \vee 1, i = 1, \dots, m \right\}, \quad (4.1)$$

де x_i – флаг, який дорівнює 1, якщо вершина a_i належить підмножині S , і який дорівнює нулю у протилежному випадку;

x – вектор стовпчик розмірності m , де x_1, \dots, x_m – його компоненти;

W – матриця ваг розмірності $m \times m$, (i, j) -й елемент якої дорівнює w_{ij} .

За допомогою напіввизначеної релаксації зведемо задачу (4.1) до задачі напіввизначеної оптимізації.

Умову $x_i = 0 \vee 1$ можна переписати в квадратичному вигляді $x_i(x_i - 1) = 0$. Як показали чисельні експерименти, у задачі напіввизначеної оптимізації цю умову задовольняє будь-який розв'язок $X = x x^T$. Тому введемо нову змінну z – вектор-стовпчик розмірності n , де z_1, \dots, z_m – компоненти вектора z , $z_i = x_i + 1$,

$x_i = z_i - 1$, та замінимо умову булевих змінних умовою $(z_i - 1)(z_i - 2) = 0$, а матрицю розв'язку будемо шукати у вигляді

$$X = (z - 1)(z - 1)^T,$$

де $(z - 1)^T = (z_1 - 1, z_2 - 1, \dots, z_m - 1)$.

Тоді перепишемо (4.1) у вигляді

$$\max \left\{ \frac{1}{2} (z - 1)^T W (z - 1) \mid \begin{array}{l} \sum_{i=1}^m z_i = k + m, \\ (z_i - 1)(z_i - 2) = 0, \quad i = 1, \dots, m \end{array} \right\}. \quad (4.2)$$

Опустимо константу та перепишемо (4.2) у вигляді

$$\min \left\{ - (z - 1)^T W (z - 1) \mid \begin{array}{l} (z_i - 1)(z_i - 2) = 0, \quad i = 1, \dots, m \\ \sum_{i=1}^m z_i = k + m \end{array} \right\}. \quad (4.3)$$

Задача (4.3) є загальною квадратичною задачею, до якої можна застосувати напіввизначену релаксацію [64].

Перепишемо (4.3) у вигляді

$$\min \{ -W \bullet Z \mid A_i \bullet Z = 0, \quad i = 1, \dots, m + 1, \quad Z \succeq 0 \}, \quad (4.4)$$

де $Z = \begin{pmatrix} 1 \\ z \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix}^T = \begin{pmatrix} 1 & z^T \\ z & z z^T \end{pmatrix}$ розмірності $(m + 1) \times (m + 1)$;

$$A_i = \begin{pmatrix} & & & i & & \\ & 2 & 0 & \dots & -1.5 & \dots & 0 \\ & 0 & \dots & & & & \\ \dots & & & & & & \\ -1.5 & 0 & \dots & 1 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & & & & & \end{pmatrix}, i=1, \dots, m;$$

$$A_{m+1} = \begin{pmatrix} (-k-m) & 0,5 & \dots & 0,5 \\ 0,5 & & & \\ \dots & & 0 & \\ 0,5 & & & \end{pmatrix}.$$

Задачу (4.4) розв'язуємо напіввизначеним симплекс-методом і отримуємо нижню оцінку цільової функції задачі (4.1). Зазначимо, що при використанні напіввизначеного симплекс-методу для того, щоб підвищити точність оцінки, потрібно додати умову виду $\bar{A} \bullet X = 1$, де $\bar{a}_{11} = 1$ та усі інші елементи матриці \bar{A} дорівнюють нулю. Це необхідно для того, щоб елемент $z_{11} = 1$.

Очевидно, що знайдені z_i будуть належати проміжку [1; 2].

Якщо знайдена матриця Z має ранг 1, тоді можна зробити висновок, що знайдений розв'язок є глобальним мінімумом задачі (4.1). Якщо ж ранг матриці Z більше 1, тоді знайдений розв'язок підставляємо у якості початкової точки в задачу (4.1) та розв'язуємо задачу (4.1) методом внутрішньої точки. Таким чином буде знайдена верхня оцінка розв'язку задачі (4.1). Чисельні експерименти показують, що у більшості випадків при використанні розв'язку напіввизначеної релаксації (4.4) у якості початкової точки, методи локального пошуку збігаються до глобального мінімуму початкової задачі (4.1).

Для знаходження нижньої оцінки цільової функції задачі (4.3) з більшою точністю можна використовувати метод точної квадратичної регуляризації [124]. Введемо нову змінну z_{n+1} . Тоді, зробивши перетворення, описані у п. 1.4.3,

отримаємо задачу

$$\max \left\{ \|y\|^2 \left| \begin{array}{l} -(\bar{y}-1)^T W(\bar{y}-1) + s + (r-1)\|y\|^2 \leq d, \\ (\bar{y}_i-1)(\bar{y}_i-2) \leq 0, \quad i=1, \dots, m, \\ -(\bar{y}_i-1)(\bar{y}_i-2) + r\|y\|^2 \leq d, \\ \sum_{i=1}^m y_i - k - m = 0. \end{array} \right. \right\},$$

$$\text{де } y = \begin{pmatrix} z_1 \\ \dots \\ z_n \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} \bar{y} \\ y_{n+1} \end{pmatrix};$$

$$\bar{y} = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} = z = \begin{pmatrix} z_1 \\ \dots \\ z_n \end{pmatrix}.$$

Визначаємо параметри s та r за формулами:

$$s \geq \|z^*\|^2 - f(z^*),$$

де z^* – розв’язок задачі (4.3), який знаходимо методом точної квадратичної регуляризації;

r – обираємо так, щоб додавання доданку $r\|z\|^2$ до неопуклих функцій задачі робило їх опуклими.

Далі знаходимо $z_{n+1} = \sqrt{f(z^*) + s - \|z^*\|^2}$ та $d = r(\|z^*\|^2 + z_{n+1}^2)$.

Отримана задача є квадратичною з опуклими квадратичними обмеженнями. Тоді до неї можна застосувати напіввизначену релаксацію [64]. Як показують чисельні експерименти, отримана таким чином нижня оцінка цільової функції знаходиться ближче до глобального мінімуму, ніж отримана нижня оцінка при

розв'язуванні задачі (4.4). Це пов'язано з тим, що допустима множина перетвореної задачі є опуклою.

4.1.2 Порівняльні чисельні експерименти

Розглянемо задачу кластеризації даних. Нехай маємо 5 вершин

$$a_1(0, 0), a_2(0, 3), a_3(2, 2), a_4(3, 0), a_5(4, 4).$$

Нам необхідно знайти підмножину S , яка включає 3 вершини, такі, що сумарна вага дуг підграфу, породженого S , є максимальною.

Знайдемо відстані між вершинами: $d_{12} = 3$, $d_{13} = 2,8284$, $d_{14} = 3$, $d_{15} = 5,6568$, $d_{23} = 2,236$, $d_{24} = 4,2426$, $d_{25} = 4,1231$, $d_{34} = 2,236$, $d_{35} = 2,8284$, $d_{45} = 4,1231$.

Нехай W – матриця ваг дуг графа G :

$$w_{ij} = w_{ji} = d_{ij}, w_{ii} = 0.$$

Запишемо отриману задачу k -klustering у вигляді задачі булевої оптимізації

$$\max \left\{ \frac{1}{2} x^T W x \mid \sum_{i=1}^n x_i = k, x_i = 0 \vee 1, i = 1, \dots, n \right\}, \quad (4.5)$$

де x_i – флаг, який дорівнює 1, якщо вершина a_i належить підмножині S , і який дорівнює нулю у протилежному випадку;

$k = 3$ та $n = 5$ з умови задачі.

Використовуючи (4.4), перепишемо (4.5) у вигляді задачі напіввизначеної оптимізації

$$\min \{ -W \bullet Z \mid A_i \cdot Z = 0, i=1, \dots, 6, Z \succeq 0 \}, \quad (4.6)$$

де $Z = \begin{pmatrix} 1 \\ z \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix}^T = \begin{pmatrix} 1 & z^T \\ z & zZ^T \end{pmatrix}$ розмірності 6×6 ;

$$A_i = \begin{pmatrix} & & & i & & \\ & 2 & 0 & \dots & -1.5 & \dots & 0 \\ & 0 & \dots & & & & \\ \dots & & & & & & \\ -1.5 & 0 & \dots & 1 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & & & & & \end{pmatrix}, i=1, \dots, 5;$$

$$A_6 = \begin{pmatrix} -8 & 0,5 & \dots & 0,5 \\ 0,5 & & & \\ \dots & & 0 & \\ 0,5 & & & \end{pmatrix}.$$

Розв'яжемо (4.6) напіввизначеним симплекс-методом. Отримали матрицю

$$\begin{pmatrix} 1,0000 & 1,6335 & 1,5956 & 1,4690 & 1,6151 & 1,6869 \\ 1,6335 & 2,9006 & 2,8428 & 2,6400 & 2,8726 & 2,9789 \\ 1,5956 & 2,8428 & 2,7867 & 2,5887 & 2,8157 & 2,9191 \\ 1,4690 & 2,6400 & 2,5887 & 2,4069 & 2,6151 & 2,7093 \\ 1,6151 & 2,8726 & 2,8157 & 2,6151 & 2,8452 & 2,9500 \\ 1,6869 & 2,9789 & 2,9191 & 2,7093 & 2,9500 & 3,0606 \end{pmatrix}$$

та нижню оцінку -41,49.

Знайдена точка дорівнює

$$z = \begin{pmatrix} 1.6335 \\ 1.5956 \\ 1.4690 \\ 1.6151 \\ 1.6869 \end{pmatrix}. \quad (4.7)$$

Для знаходження верхньої оцінки розв'язку будемо розв'язувати початкову задачу (4.5) методом внутрішньої точки, та у якості початкової точки візьмемо точку (4.7). Знайдений розв'язок

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

є оптимальним (значення цільової функції дорівнює -25,56).

На рис. 4.1 графічно зображено розв'язок поставленої задачі.

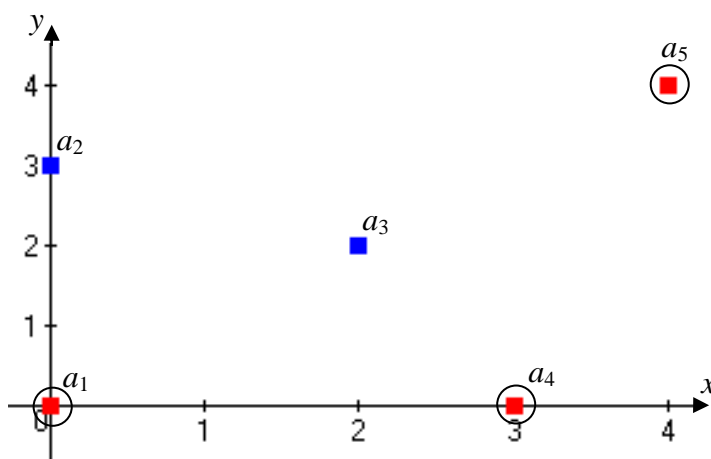


Рисунок 4.1 – Розв'язок задачі кластеризації даних, де

■ – точки, які не увійшли в кластер;

⊙ – точки, які увійшли в кластер.

4.2 Модель розміщення сенсорних датчиків у мережі

4.2.1 Постановка задачі і метод розв'язування

Важливою задачею, де використовується напіввизначена релаксація, є локалізація (розміщення) датчиків у мережі. Останніми роками ця проблема інтенсивно досліджується [6, 13, 22, 38, 45, 53, 62, 66, 86, 92]. Важливість цієї задачі полягає в тому, що в сучасному інформаційному суспільстві існує проблема автоматичного збору даних. Одним з основних чинників автоматизації збору даних є датчики, які забезпечують інформаційні мережі даними. Від кількості та структури датчиків залежить ефективність обробки даних, які поступають у комп'ютер. Існують системи, які вже мають якусь кількість датчиків із заданою структурою, і необхідно доповнити цю структуру новими датчиками. Задача локалізації датчиків у мережі полягає в тому, щоб визначити таку структуру датчиків, яка повністю охоплюватиме заданий об'єкт. При великій кількості датчиків задача стає складною та потребує розробки нових методів її розв'язування.

Нехай маємо граф $G = (V, E)$ та набір невід'ємних ваг $\{d_{ij} : (i, j) \in E\}$. В задачі локалізації датчиків у мережі вершини графа G позначають датчики, дуги графа G позначають комунікаційні шляхи, а ваги – відстані. До того ж, вершини розділені на дві підмножини: перша підмножина – множина закріплених вершин, чия точна позиція відома, друга підмножина – множина датчиків, чие розташування невідомо. Метою є визначити позицію усіх нових датчиків. В [53, 66, 86, 105] представлений наближений поліноміальний алгоритм для розв'язування цієї задачі, який використовує двоїсту теорію та метод внутрішньої точки для SDP.

Розглянемо n -вимірний евклідовий простір. Нехай маємо m закріплених вершин графа $a^1, \dots, a^m \in R^d$ та k вершин $x^1, \dots, x^k \in R^d$, чие розташування нам необхідно визначити. До того ж, нам відомі p значень відстаней d_{ij} між a^i та x^j для деяких i, j , та l значень \bar{d}_{ij} між x^i та x^j для деяких $i < j$. Введемо

позначення: нехай $N_a = \{(i, j) : d_{ij} \text{ визначено}\}$ та $N_x = \{(i, j) : i < j, \bar{d}_{ij} \text{ визначено}\}$. Задача локалізації датчиків у мережі полягає у пошуку таких $x^1, \dots, x^k \in R^d$, які задовольняють наступним умовам:

$$\begin{aligned} \|a^i - x^j\|^2 &= d_{ij}^2 \quad \forall (i, j) \in N_a, \\ \|x^i - x^j\|^2 &= \bar{d}_{ij}^2 \quad \forall (i, j) \in N_x. \end{aligned} \quad (4.8)$$

Таким чином, задача локалізації датчиків у мережі зводиться до розв'язування нелінійної квадратичної системи рівнянь (4.8). Розв'язати цю задачу достатньо складно, тому замінимо її оптимізаційною задачею

$$\min \left\{ \|x\|^2 \left| \begin{array}{l} \|a^i - x^j\|^2 = d_{ij}^2, \quad \forall (i, j), i=1, \dots, m, j=1, \dots, k \\ \|x^i - x^j\|^2 = \bar{d}_{ij}^2, \quad i < j, i=1, \dots, k, j=2, \dots, k \end{array} \right. \right\}. \quad (4.9)$$

Задача (4.9) може не завжди мати розв'язок, тому будемо розв'язувати наступну задачу

$$\min \left\{ \|u\|^2 + \|v\|^2 \left| \begin{array}{l} \|a^i - x^j\|^2 = d_{ij}^2 + u_{ij}, \quad \forall (i, j), i=1, \dots, m, j=1, \dots, k \\ \|x^i - x^j\|^2 = \bar{d}_{ij}^2 + v_{ij}, \quad i < j, i=1, \dots, k, j=2, \dots, k \end{array} \right. \right\}, \quad (4.10)$$

де u_{ij} – відхилення від заданої відстані між i -тою вершиною та j -тим датчиком ($\forall (i, j), i=1, \dots, m, j=1, \dots, k$);

v_{ij} – відхилення від заданої відстані між i -тим та j -тим датчиками ($i < j, i=1, \dots, k, j=2, \dots, k$).

Усі значення u_{ij} , де $i=1, \dots, m$, $j=1, \dots, k$, формують вектор u розмірності p , де p – кількість відомих відстаней d_{ij} . Усі значення v_{ij} , де $i < j$, $i=1, \dots, k$, $j=2, \dots, k$, формують вектор v розмірності l , де l – кількість відстаней \bar{d}_{ij} .

Таким чином, цільова функція задачі (4.10) мінімізує відхилення від заданих відстаней. Очевидно, що задача (4.10) завжди має розв'язок.

Цю квадратичну оптимізаційну задачу (4.10) перетворимо на задачу напіввизначеної оптимізації та використаємо напіввизначену релаксацію для її розв'язування. У більшості ранніх методів дана проблема зводилась до задачі нелінійної глобальної оптимізації. Альтернативний підхід – з використання SDP – був розроблений відносно недавно [66]. Удосконалимо релаксацію, яка використовується в [66], для нашої постановки задачі (4.10), яка враховує відхилення від заданих відстаней. Отримаємо наступну задачу напіввизначеної оптимізації

$$\min \left\{ C \bullet X \left| \begin{array}{l} A_{ij} \bullet X = 0, \quad \forall(i, j), \quad i=1, \dots, m, \quad j=1, \dots, k, \quad X \succeq 0 \\ \bar{A}_{ij} \bullet X = 0, \quad i < j, \quad i=1, \dots, k, \quad j=2, \dots, k, \quad X \succeq 0 \end{array} \right. \right\}, \quad (4.11)$$

$$\text{де } X = \begin{pmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{pmatrix} \text{ розмірності } (n+k+p+l) \times (n+k+p+l); \quad (4.12)$$

$$X_1 = I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & 1 \end{pmatrix} \text{ розмірності } n \times n;$$

$$X_2 = \begin{pmatrix} x_1^1 & \dots & x_1^k & u_1 & \dots & u_p & v_1 & \dots & v_l \\ \vdots & \ddots & \vdots & 0 & \dots & 0 & 0 & \dots & 0 \\ x_n^1 & \dots & x_n^k & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \text{ розмірності } n \times (k+p+l);$$

$X_3 = X_2^T X_2$ розмірності $(k + p + l) \times (k + p + l)$.

Очевидно, що матриця C буде мати наступний вигляд

$$C = \begin{pmatrix} 0 \dots 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 \dots 0 & 0 \\ 0 \dots 0 & I \end{pmatrix}, \quad (4.13)$$

де I – одинична матриця розмірності $(p + l) \times (p + l)$.

Розрахуємо матриці обмежень A_{ij} . З (4.10) маємо

$$\begin{aligned} & \|a^i - x^j\|^2 - d_{ij}^2 - u_{ij} = -d_{ij}^2 + \|a^i\|^2 + \|x^j\|^2 - 2a^i x^j - u_{ij} = \\ & = -d_{ij}^2 + (a_1^{i^2} + \dots + a_n^{i^2}) + (x_1^{j^2} + \dots + x_n^{j^2}) - 2(a_1^i x_1^j + \dots + a_n^i x_n^j) - u_{ij} = 0. \end{aligned} \quad (4.14)$$

Враховуючи (4.12) та (4.14), маємо, що

$$A_{ij} = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_j' \end{pmatrix} \text{ розмірності } (n + k + p + l) \times (n + k + p + l), \quad (4.15)$$

$$A_1 = \begin{pmatrix} (-d_{ij}^2 + \|a^i\|^2) & 0 \dots 0 \\ 0 & 0 \dots 0 \\ \vdots & \ddots \\ 0 & 0 \dots 0 \end{pmatrix} \text{ розмірності } n \times n,$$

$$A_2 = \begin{pmatrix} & j & & & & u_{ij} \\ 0 & \dots & -a_1^i & 0 & \dots & -0.5 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & -a_n^i & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \text{ розмірності } n \times (k + p + l),$$

$$A'_j = \begin{pmatrix} & & & j & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & \dots & & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & \dots & & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \text{розмірності } (k+p+l) \times (k+p+l).$$

Тобто в матриці A_2 у j -тому стовпці елементи дорівнюють компонентам вектора a^i , взятим з протилежним знаком, та елемент, відповідний u_{ij} в матриці X_2 , дорівнює -0.5 . Всі елементи матриці A'_j дорівнюють нулю, окрім $a_{jj} = 1$.

Розрахуємо матриці обмежень \bar{A}_{ij} . З (4.10) маємо

$$\begin{aligned} & \|x^i - x^j\|^2 - \bar{d}_{ij}^2 - v_{ij} = -\bar{d}_{ij}^2 + \|x^i\|^2 + \|x^j\|^2 - 2x^i x^j - v_{ij} = \\ & = -\bar{d}_{ij}^2 + (x_1^{i^2} + \dots + x_n^{i^2}) + (x_1^{j^2} + \dots + x_n^{j^2}) - 2(x_1^i x_1^j + \dots + x_n^i x_n^j) - v_{ij} = 0. \end{aligned} \quad (4.16)$$

Тоді, враховуючи (4.12) та (4.16), маємо, що

$$\bar{A}_{ij} = \begin{pmatrix} \bar{A}_1 & \bar{A}_2 \\ \bar{A}_2^T & \bar{A}'_{ij} \end{pmatrix} \text{розмірності } (n+k+p+l) \times (n+k+p+l), \quad (4.17)$$

$$\bar{A}_1 = \begin{pmatrix} -\bar{d}_{ij}^2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & 0 \end{pmatrix} \text{розмірності } n \times n,$$

$$\bar{A}_2 = \begin{pmatrix} & & & v_{ij} & & & \\ 0 & \dots & 0 & -0.5 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & 0 & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \text{розмірності } n \times (k + p + l),$$

$$\bar{A}'_{ij} = \begin{pmatrix} & i & j & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & 1 & 0 & -1 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & -1 & 0 & 1 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \text{розмірності } (k + p + l) \times (k + p + l).$$

Тобто в матриці A_2 усі елементи дорівнюють нулю, окрім елемента, відповідного v_{ij} в матриці X_2 , який дорівнює -0.5 . Всі елементи матриці A'_j дорівнюють нулю, окрім $a_{ii} = a_{jj} = 1$, $a_{ij} = a_{ji} = -1$.

Очевидно, що кількість обмежень A_j буде p , а кількість обмежень \bar{A}'_{ij} буде l .

Таким чином, задачу (4.11) можемо записати в загальному вигляді:

$$\min \{ C \bullet X \mid A_i \bullet X = b_i, X \succeq 0 \}. \quad (4.18)$$

Напіввизначена релаксація передбачає релаксацію обмеження на ранг матриці X у задачі (4.18). Задача (4.18) може бути розв'язана методом внутрішньої точки [82] або напіввизначеним симплекс-методом [125].

4.2.2 Порівняльні чисельні експерименти

В наступних прикладах задача локалізації датчиків у мережі за допомогою напіввизначеної релаксації зводилася до задачі напіввизначеної

оптимізації. Задача напіввизначеної оптимізації розв'язувалась напіввизначеним симплекс-методом [127].

Експеримент 1 (дві відомі вершини, один датчик)

$$a_1 (1, 0),$$

$$a_2 (-1, 0),$$

$$d_{11} = d_{12} = \sqrt{2},$$

$$x_1 - ?$$

Напіввизначений симплекс-метод знайшов розв'язок $x_1 (0, 1)$. Графічно положення відомих та знайденої вершини представлено на рис. 4.2.

Розмірність матриці X у перетвореній задачі: 3×3 .

Ранг X дорівнює 1 \Rightarrow знайдений оптимальний розв'язок задачі.

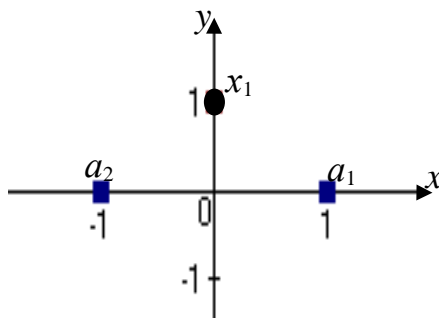


Рисунок 4.2 – Положення датчиків, експеримент 1, де

■ – відома вершина;

● – датчик.

Експеримент 2 (чотири відомі вершини, чотири датчика)

$$a_1 (1, 1),$$

$$a_2 (1, -1),$$

$$a_3 (-1, -1),$$

$$a_4 (-1, 1),$$

$$d_{a_1x_1} = d_{a_2x_2} = d_{a_3x_3} = d_{a_4x_4} = 1,$$

$$d_{x_1x_2} = d_{x_2x_3} = d_{x_3x_4} = d_{x_4x_1} = 2 - \sqrt{2},$$

$$x_1, x_2, x_3, x_4 - ?$$

Розмірність матриці X у перетвореній задачі: 9×9 . Графічно положення відомих та знайденої вершини представлено на рис. 4.3.

Ранг X дорівнює 1 \Rightarrow знайдений оптимальний розв'язок.

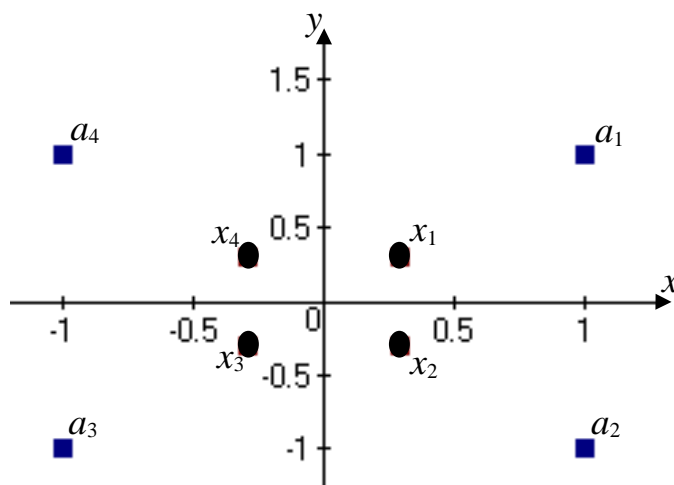


Рисунок 4.3 – Положення датчиків, експеримент 2, де

■ – відома вершина;

● – датчик.

Експеримент 3

Розглянемо задачу локалізації датчиків у мережі з трьома відомими вершинами та двома датчиками. Маємо відомі вершини

$$a_1(1, 0), a_2(2, 1), a_3(0, 3),$$

задані відстані між відомими вершинами та датчиками

$$d_{11} = d_{21} = d_{31} = d_{12} = 1,$$

та відстань між датчиками

$$\bar{d}_{12} = 1.$$

Графічно положення відомих вершин представлено на рис. 4.4. З рис. 4.4 очевидно, що задача (4.9) не має розв'язку (неможливо розташувати датчики так, щоб відстані між ними та відомими вершинами задовольняли задані відстані).

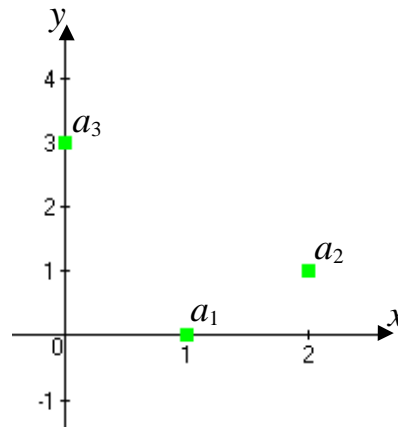


Рисунок 4.4 – Задані вершини

Тому будемо розв'язувати задачу (4.10), яка має наступний вигляд

$$\min \left\{ \|u\|^2 + \|v\|^2 \left| \begin{array}{l} \|a^i - x^j\|^2 = d_{ij}^2 + u_{ij}, \quad \forall(i, j), \quad i = 1, \dots, m, \quad j = 1, \dots, k \\ \|x^i - x^j\|^2 = \bar{d}_{ij}^2 + v_{ij}, \quad i < j, \quad i = 1, \dots, k, \quad j = 2, \dots, k \end{array} \right. \right\}. \quad (4.19)$$

З умови задачі слідує, що $m = 3$, $k = 2$, $n = 2$, $p = 4$, $l = 1$. Перетворимо задачу (4.19) на задачу напіввизначеної оптимізації. Для цього використаємо схему (4.11), (4.12), (4.13), (4.15), (4.17).

Отримаємо наступну задачу напіввизначеної оптимізації

$$\min \left\{ C \bullet X \left| \begin{array}{l} A_{ij} \bullet X = 0, \quad \forall(i, j), \quad i = 1, \dots, m, \quad j = 1, \dots, k, \quad X \succeq 0 \\ \bar{A}_{ij} \bullet X = 0, \quad i < j, \quad i = 1, \dots, k, \quad j = 2, \dots, k, \quad X \succeq 0 \end{array} \right. \right\}, \quad (4.20)$$

$$\text{де } C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, A_{11} = \begin{pmatrix} 0 & 0 & -1 & 0 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

$$A_{21} = \begin{pmatrix} 4 & 0 & -2 & 0 & 0 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_{31} = \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

$$A_{12} = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \bar{A}_{12} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Вхідні дані для програми в Ехсел знаходяться в додатку Б, таблиця Б.2. Зауважимо, що для розв'язування задачі (4.20) напіввизначеним симплекс-методом додаються ще три обмеження (див. три останні обмеження у таблиці

Б.2), які задають умову того, що матриця X_1 (див. 4.12) повинна бути одиничною.

Симплекс-метод знайшов наближений розв'язок задачі (4.20), який дорівнює 5,6485. Матриця розв'язку X має ранг 3, а це означає, що ми знайшли наближений розв'язок початкової задачі (4.19)

$$X = \begin{pmatrix} 1,0000 & 0,0000 & 0,7612 & 1,3523 & 1,3914 & 0,8133 & 1,7466 & 0,0001 & 0,0000 \\ 0,0000 & 1,0000 & 1,5279 & 0,8366 & 0,0077 & -0,0002 & 0,0009 & 0,0001 & 0,0002 \\ 0,7612 & 1,5279 & 2,9138 & 2,3093 & 1,0710 & 0,6188 & 1,3308 & 0,0001 & 0,0002 \\ 1,3523 & 0,8366 & 2,3093 & 2,7047 & 1,8888 & 1,0997 & 2,3629 & 0,0001 & 0,0001 \\ 1,3914 & 0,0077 & 1,0710 & 1,8888 & 1,9363 & 1,1317 & 2,4303 & 0,0001 & 0,0000 \\ 0,8133 & -0,0002 & 0,6188 & 1,0997 & 1,1317 & 0,6615 & 1,4206 & 0,0001 & 0,0000 \\ 1,7466 & 0,0009 & 1,3308 & 2,3629 & 2,4303 & 1,4206 & 3,0507 & 0,0001 & -0,0001 \\ 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0001 & 0,0000 & 0,0000 \\ 0,0000 & 0,0002 & 0,0002 & 0,0001 & 0,0000 & 0,0000 & -0,0001 & 0,0000 & 0,0000 \end{pmatrix}. \quad (4.21)$$

З матриці (4.21) отримаємо наближений розв'язок задачі(4.19)

$$x_1 = \begin{pmatrix} 0,7612 \\ 1,5279 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 1,3523 \\ 0,8366 \end{pmatrix},$$

та наближені значення відхилень

$$u_{11} = 1,3914, \quad u_{12} = 0,8133, \quad u_{13} = 1,7466, \quad u_{21} = 0,0001, \quad v_{12} = 0.$$

Для отримання верхньої оцінки використаємо знайдені вектори x , u та v в якості початкових точок та будемо розв'язувати задачу (4.19) методом внутрішньої точки. Отримаємо координати датчиків

$$x_1 = \begin{pmatrix} 0,7634 \\ 1,5297 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 1,5075 \\ 0,8616 \end{pmatrix},$$

при відхиленнях

$$u_{11} = 1,396, u_{12} = 0,8097, u_{13} = 1,7445, u_{21} = 0, v_{12} = 0,$$

та значення мінімуму 5,6479, яке є оптимальним.

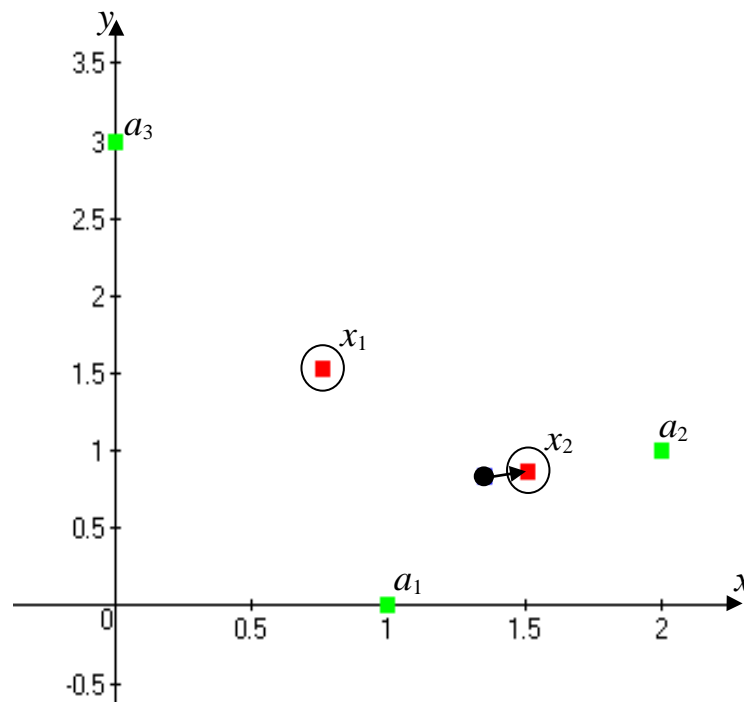


Рисунок 4.5 – Оптимальне розташування датчиків, де

- – відомі вершини;
- – положення другого датчика, знайдене за допомогою напіввизначеної релаксації (положення першого датчика співпало з оптимальним положенням);
- ⊙ – оптимальне розташування датчиків (знайдена верхня оцінка розв’язку початкової задачі).

На рис. 4.5 зображені положення початкових точок, положення датчиків, знайдених за допомогою напіввизначеної релаксації початкової задачі, та оптимальне розташування датчиків, знайдене шляхом розв’язання початкової задачі методом внутрішньої точки, де в якості початкових точок бралися нижні оцінки, знайдені за допомогою напіввизначеної релаксації. Як можна побачити з рис.4.5, за допомогою напіввизначеної релаксації було знайдене оптимальне положення першого датчика та наближене положення

другого датчика, а отримана верхня оцінка задачі виявилася глобальним мінімумом цієї задачі.

4.3 Задача Max-Cut

4.3.1 Постановка задачі і метод розв'язування

Пошук максимального розрізу графа (в англійській літературі ця проблема називається max-cut) є однією зі стандартних NP-складних задач комбінаторної оптимізації та найпростішою з задач розфарбування графа [3, 5, 31, 35, 43, 54, 56, 57, 74, 75, 88, 90, 96, 107, 110].

Розглянемо неорієнтований граф G з вершинами $N = \{1, \dots, n\}$ та множиною дуг E . Нехай $w_{ij} = w_{ji}$ – вага дуги $(i, j) \in E$. Будемо вважати, що $w_{ij} \geq 0$ для усіх $(i, j) \in E$. Задачею max-cut називається задача визначення підмножини S вершин N , для якої сума ваг дуг, що йдуть з S в \bar{S} , максимальна ($\bar{S} = N \setminus S$).

Нехай $x_j = 1$ для $j \in S$ та $x_j = -1$ для $j \in \bar{S}$. Тоді задача max-cut формулюється як задача цілочисельної оптимізації:

$$\max \left\{ \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (1 - x_i x_j) \mid x_j \in \{-1, 1\}, j = 1, \dots, n \right\}. \quad (4.22)$$

Введемо позначення: $Y = xx^T$, W – матриця, (i, j) -й елемент якої рівний w_{ij} . Тоді задача (4.22) може бути записана у вигляді:

$$\max \left\{ \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n w_{ij} - \frac{1}{4} W \bullet Y \mid Y = xx^T, \text{diag}(Y) = 1, \text{rank}(Y) = 1, j = 1, \dots, n \right\}. \quad (4.23)$$

Якщо ми в (4.23) не будемо враховувати обмеження на ранг матриці (напіввизначена релаксація), то отримаємо наступну задачу напіввизначеної

оптимізації:

$$\max \left\{ \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n w_{ij} - \frac{1}{4} W \bullet Y \mid Y \succeq 0, \text{diag}(Y) = 1, j = 1, \dots, n \right\},$$

яку можна переписати у вигляді:

$$\min \{ W \bullet Y \mid Y \succeq 0, \text{diag}(Y) = 1 \}. \quad (4.24)$$

Задача (4.24) розв'язувалась за допомогою напіввизначеного симплекс-методу [125].

Для знаходження кращої нижньої оцінки цільової функції задачі (4.22) перепишемо її у вигляді

$$\min \left\{ \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j \mid x_j (x_j - 1) = 0, j = 1, \dots, n \right\}. \quad (4.25)$$

Тоді до задачі (4.25) можна застосувати метод точної квадратичної регуляризації [124]. Введемо нову змінну x_{n+1} . Тоді, зробивши перетворення, описані у п. 1.4.3, отримаємо задачу

$$\max \left\{ \|z\|^2 \mid \begin{array}{l} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \bar{z}_i \bar{z}_j + s + (r-1) \|z\|^2 \leq d, \\ \bar{z}_j (\bar{z}_j - 1) \leq 0, \quad j = 1, \dots, n, \\ -\bar{z}_j (\bar{z}_j - 1) + r \|z\|^2 \leq d. \end{array} \right\},$$

$$\text{де } z = \begin{pmatrix} z_1 \\ \dots \\ z_n \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} \bar{z} \\ z_{n+1} \end{pmatrix};$$

$$\bar{z} = \begin{pmatrix} z_1 \\ \dots \\ z_n \end{pmatrix} = x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}.$$

Визначаємо параметри s та r за формулами:

$$s \geq \|x^*\|^2 - f(x^*),$$

де x^* – розв’язок задачі (4.22) методом точної квадратичної регуляризації;
 r – обираємо так, щоб додавання доданку $r\|x\|^2$ до неопуклих функцій задачі робило їх опуклими.

Далі знаходимо

$$x_{n+1} = \sqrt{f(x^*) + s - \|x^*\|^2}, \quad d = r(\|x^*\|^2 + x_{n+1}^2).$$

Отримана задача є квадратичною. Тоді до неї можна застосувати напіввизначену релаксацію [64]. Як показують чисельні експерименти, отримана таким чином нижня оцінка є кращою, ніж отримана нижня оцінка при розв’язуванні задачі (4.23).

4.4 Програмне забезпечення для розв’язування задач напіввизначеної оптимізації

Для написання програм для розв’язування задач напіввизначеної оптимізації була обрана мова VBA та середовище Excel з наступних причин:

- зручність середовища Excel для завдання вхідних даних – матриць;
- мова VBA вбудована в прикладення Office, і код на мові VBA можна зберігати усередині документів Office, у нашому випадку, в книзі Excel, та з зручністю викликати його прямо в документі подвійним клацанням миші;
- при використанні даної програми користувачеві не доведеться піклуватися про установку та настройку спеціального середовища програмування та про наявність потрібних бібліотек — MS Office є практично на будь-якому комп'ютері;
- програмне забезпечення орієнтовано на використання в навчальному процесі студентами університетів для виконання лабораторних, курсових та дипломних робіт.

Розглянемо задачу напіввизначеного програмування:

$$\min \{ C \bullet X \mid A_i \bullet X = b_i, \quad i=1, \dots, m, \quad X \succeq 0 \},$$

де C – симетрична матриця ($n \times n$);

A_i – симетричні матриці ($n \times n$);

b – вектор розмірності m ;

X – симетрична додатно напіввизначена матриця ($n \times n$).

Вхідними даними в програмі є матриця цільової функції C , нижче – матриці обмежень A_i та праворуч – відповідний елемент вектора b_i . Вихідні дані програми – знайдена матриця X та значення цільової функції $C \bullet X$. Значення матриці X виводиться рядком нижче від заданих матриць A_i . Значення цільової функції виводиться праворуч від матриці X . Якщо ж розв'язок задачі необмежений або допустима множина є пустою, то видається відповідне повідомлення.

На рис. 4.6 представлена схема алгоритму напіввизначеного симплекс-методу, описаного у п. 2.2, для розв'язування задач SDP. На рис. 4.7 представлена схема алгоритму методу внутрішньої точки з допустимою початковою точкою, описаного у п. 1.2, для розв'язування задач SDP.

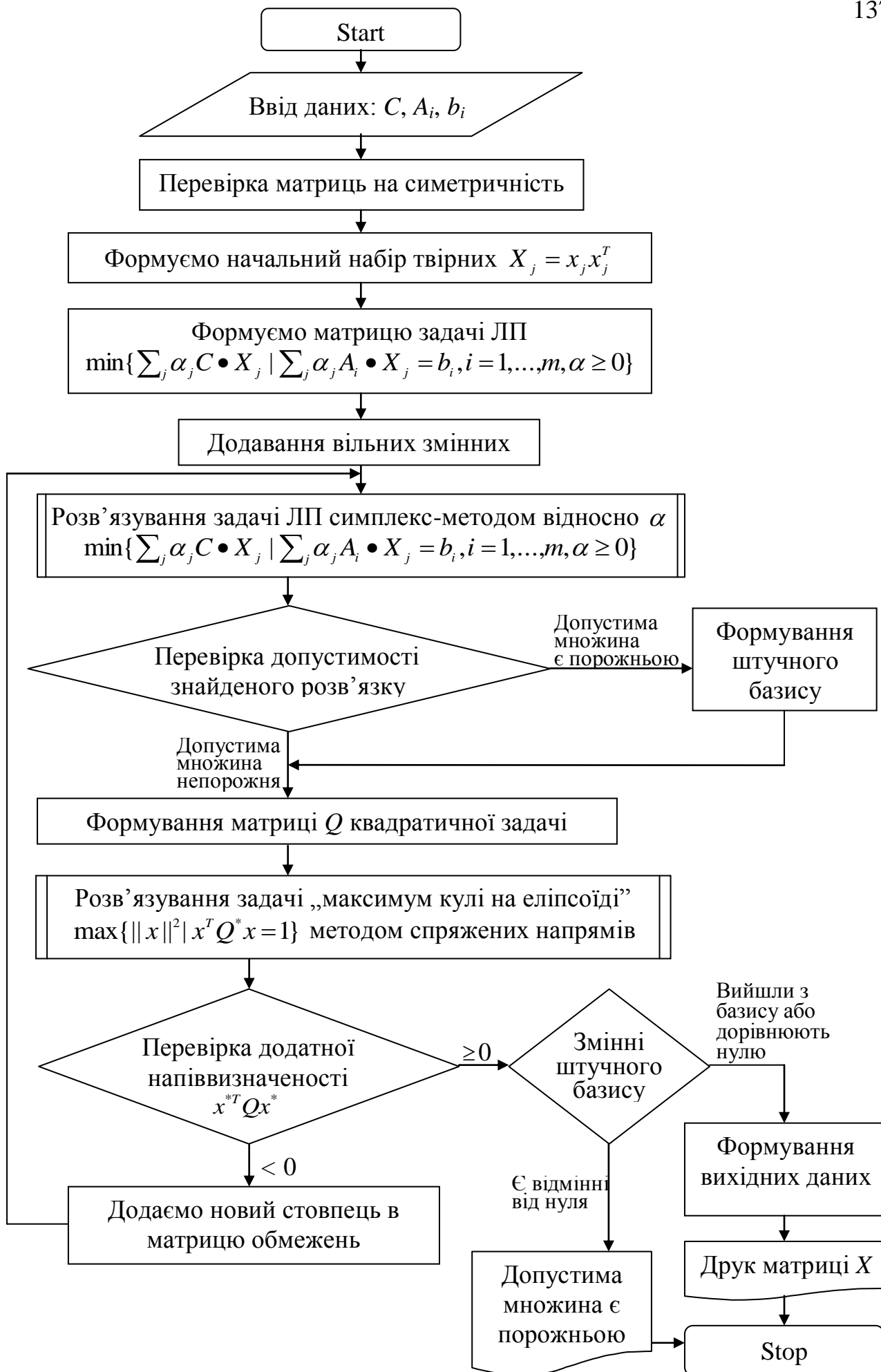


Рисунок 4.6 – Схема алгоритму напіввизначеного симплекс-методу для розв'язання задач SDP

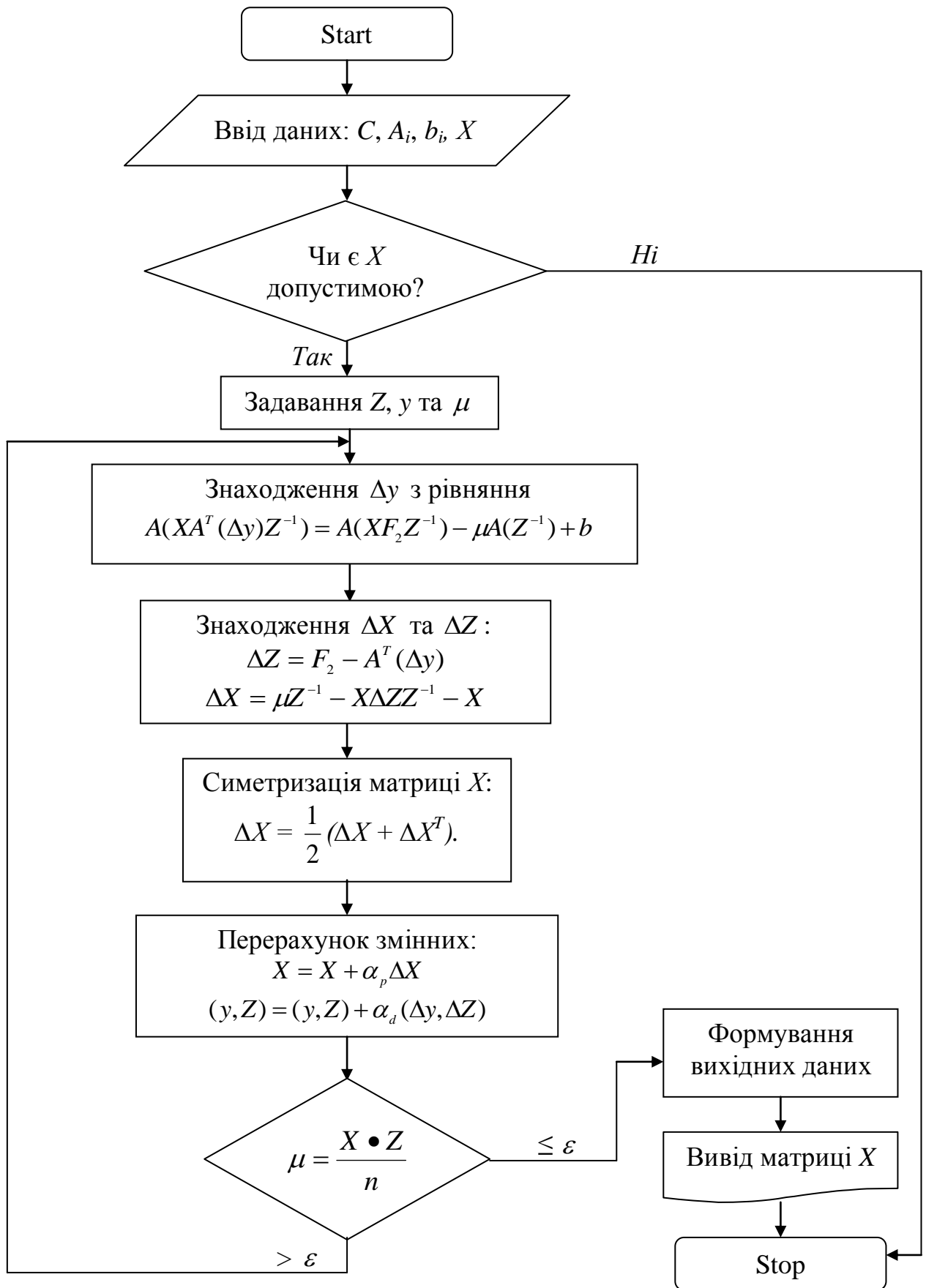


Рисунок 4.7 – Схема алгоритму методу внутрішньої точки з допустимою початковою точкою для розв'язання задач SDP

4.5 Висновки розділу

Основними результатами розділу є:

– описано та удосконалено використання напіввизначеної релаксації для широкого кола прикладних задач напіввизначеного програмування: задачі кластеризації даних, задачі розміщення сенсорних датчиків у мережі та задачі пошуку максимального розрізу графа. Проведені чисельні експерименти для цих класів задач підтвердили ефективність обраного методу;

– розраховано напіввизначену релаксацію розширеної постановки задачі локалізації датчиків у мережі з урахуванням відхилень, що дало змогу знаходити розв’язки задач, які у стандартній постановці розв’язку не мають. Проведені чисельні експерименти для цього класу задач підтвердили ефективність обраного методу;

– для уточнення нижньої оцінки цільової функції в задачах кластеризації даних і розміщення датчиків у мережі, ці задачі вперше були перетворені на задачу максимізації норми вектора на перетині еліпсоїдів, що дало змогу знаходити кращу нижню оцінку цільової функції початкової задачі (це пов’язано з тим, що допустима множина перетвореної задачі є опуклою);

– розглянута техніка використання напіввизначеної релаксації для даних задач може бути використана також для широкого кола інших прикладних задач, математичні моделі яких можуть бути представлені загальними квадратичними функціями.

Основні результати розділу опубліковано в [126, 127, 130, 133, 138–140, 152, 153].

ВИСНОВКИ

У дисертаційній роботі розв'язуються задачі напіввизначеної оптимізації та загальні квадратичні задачі, які мають значне застосування в математичному моделюванні складних систем. Основні наукові результати дисертації такі:

– удосконалено та обґрунтовано напіввизначений симплекс-метод як альтернативу існуючим прямо-двоїстим методам внутрішньої точки для розв'язування задач напіввизначеної оптимізації, що дало можливість розширити межі ефективного використання математичного моделювання складних систем;

– уперше використано метод спряжених напрямів у процедурі оберненої ітерації для визначення додатної напіввизначеності матриць (шляхом знаходження власного вектора, відповідного мінімальному власному значенню), що за результатами чисельних експериментів дозволило підвищити точність розрахунків та прискорити збіжність до власного вектора матриці;

– для розв'язування задач напіввизначеної оптимізації удосконалено напіввизначений симплекс-метод та прискорено його збіжність шляхом використання процедури оберненої ітерації та методу спряжених напрямів;

– уперше строго доведено збіжність напіввизначеного симплекс-методу за допомогою теорем і лем;

– визначено теоретичні переваги напіввизначеного симплекс-методу над прямо-двоїстими методами внутрішньої точки, наведено задачі, для яких напіввизначений симплекс-метод знаходить розв'язки, а метод внутрішньої точки не знаходить;

– проведено порівняння результатів програми, яка реалізує напіввизначений симплекс-метод, з відомими програмними пакетами, які реалізують методи внутрішніх точок. Ці експерименти показали, що дана програмна реалізація напіввизначеного симплекс-методу не поступається

популярним програмним пакетам методів внутрішньої точки навіть при тому, що його програмна реалізація не є оптимальною;

– уперше використано точну квадратичну регуляризацію для перетворення загальних квадратичних задач до задач максимізації норми вектора на перетині еліпсоїдів, що дало змогу знаходити кращу нижню оцінку цільової функції початкової задачі. Для уточнення нижньої оцінки цільової функції в задачах кластеризації даних і задачах локалізації датчиків у мережі, ці задачі вперше були перетворені на задачу максимізації норми вектора на перетині еліпсоїдів;

– удосконалено напіввизначену релаксацію для квадратичних задач з булевими змінними та прикладних задач, математичні моделі яких описуються з використанням загальних квадратичних функцій (задачі кластеризації даних, задачі розміщення датчиків у мережі, задачі пошуку максимального розрізу графа) шляхом модифікації обмежень задачі. Це дозволило знайти кращі нижні оцінки цільової функції задачі. Розглянута техніка використання напіввизначеної релаксації для даних задач може бути використана також для широкого кола інших прикладних задач, математичні моделі яких можуть бути представлені загальними квадратичними функціями;

– введено нове поняття точності напіввизначеної релаксації та для окремих класів задач максимізації норми вектора на опуклій множині доведена точність напіввизначеної релаксації;

– розраховано напіввизначену релаксацію розширеної постановки задачі локалізації датчиків у мережі з урахуванням відхилень, що дало змогу знаходити розв'язки задач, які у стандартній постановці розв'язку не мають;

– уперше запропоновано та перевірено на практиці нову процедуру для знаходження верхніх і нижніх оцінок цільової функції у загальних задачах квадратичної оптимізації. Як показали чисельні експерименти, в 91,66 % випадків отримана верхня оцінка співпадала з глобальним мінімумом

початкової задачі.

Виконані дослідження дозволяють зробити висновок, що подальше вдосконалення напіввизначеної релаксації пов'язано з використанням точної квадратичної регуляризації в напіввизначеній оптимізації, яка дозволяє при визначених умовах отримувати значення глобального мінімуму загальних квадратичних задач з більшою точністю. Крім цього, подальші дослідження будуть спрямовані на розширення класів задач, для яких напіввизначена релаксація буде точною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agrawal M. Camera calibration using spheres: A semidefinite programming approach [Електронний ресурс] / М. Agrawal, L. S. Davis. – Електрон. дані (1 файл). – University of Maryland, 2007. – 8 р. – Режим доступу: www.ai.sri.com/~agrawal/ICCV03_camera_calibration.pdf.
2. Agrawal M. On automatic determination of varying focal lengths using semidefinite programming [Електронний ресурс] / М. Agrawal. – Електрон. дані (1 файл). – Menlo Park, 2004. – 4 р. – Режим доступу: www.ai.sri.com/~agrawal/icip04_selfcalibration.pdf.
3. Alizadeh F. Interior point methods in semidefinite programming with applications to combinatorial optimization [Електронний ресурс] / F. Alizadeh. – Електрон. дані (1 файл). – Rutgers University, 1993. – 40 р. – Режим доступу: rutcor.rutgers.edu/~alizadeh/MYPAPERS/sdp.ps.gz.
4. Alizadeh F. Primal-Dual Interior Point Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results [Електронний ресурс] / F. Alizadeh, J.-P. A. Haeberly, M. L. Overton. – Електрон. дані (1 файл). – SIAM, 1998. – 23 р. – Режим доступу: www.cs.nyu.edu/~overton/papers/pdf/pdsdp.pdf.
5. An Interior-Point Method for Semidefinite Programming [Електронний ресурс] / Helmborg C., Rendl F., Vanderbei R.J., Wolkowicz H. – Електрон. дані (1 файл). – Princeton: Princeton University, 1994. – 24 р. – Режим доступу: <https://www-user.tu-chemnitz.de/~helmborg/hrvw.pdf>.
6. Anderson B.D.O. Wireless sensor network localization techniques / B.D.O. Anderson, G. Mao, B. Fidan // Computer Networks. – 2007. – № 51. – Р. 2529–2553.
7. Antoniou A. Chapter 14. Semidefinite and Second-Order Cone Programming [Електронний ресурс] / A. Antoniou, W.-S. Lu // Practical Optimization. – Електрон. дані (1 файл). – USA: Springer. – Р. 449–500. – Режим доступу: shmathsoc.org.cn/lu/core%2520part/Chap14.pdf.
8. Bastos F. An Extension of a Variant of Predictor-Corrector Primal-Dual

Method from Linear Programming to Semidefinite Programming [Электронный ресурс] / F. Bastos, A. Teixeira. – Электрон. дані (1 файл). – 2005. – 24 р. – Режим доступу: www.scielo.oces.mctes.pt/pdf/iop/v25n2/v25n2a06.

9. Bellman R. On systems of linear inequalities in Hermitian matrix variables / R. Bellman, K. Fan // In Proc. Sympos. Pure Math. – 1963. – Vol. VII. – P. 1–11.

10. Benson S.J. DSDP5: Software for Semidefinite Programming [Электронный ресурс] / S.J. Benson, Y. Ye. – Электрон. дані (1 файл). – ACM Transactions on Mathematical Software, 2005. – 21 р. – Режим доступу: web.stanford.edu/~yye/DSDP5-Paper.pdf.

11. Benson S.J. Parallel Semidefinite programming and combinatorial optimization / S.J. Benson // Parallel combinatorial optimization. – New Jersey: Wiley-interscience, 2006. – P. 103–122.

12. Benterki D. A numerical implementation of an interior point method for semidefinite programming [Электронный ресурс] / D. Benterki, J. P. Crouzeix, B. Merikhi. – Электрон. дані (1 файл). – Rio de Janeiro, 2003. – 59 р. – Режим доступу: www.scielo.br/pdf/pepe/v23n1/a05v23n1.

13. Biswas P. Semidefinite programming for ad hoc wireless sensor network localization [Электронный ресурс] / P. Biswas, Y. Ye. // In Proceedings of the 3-rd International Symposium on Information Processing in Sensor Networks. – Электрон. дані (1 файл). – Berkeley, CA, USA, 2004. – P. 46–54. – Режим доступу: web.stanford.edu/~yye/adhocn4.pdf.

14. Blekherman G. Semidefinite Optimization and Convex Algebraic Geometry [Электронный ресурс] / Grigoriy Blekherman, Pablo A. Parrilo, Rekha R. Thomas. – Электрон. дані (1 файл). – SIAM, 2013. – 476 р. – Режим доступу: <http://www.math.washington.edu/~thomas/frg/frgbook/SIAMBookFinalvNov12-2012.pdf>.

15. Bomze I. M. Solving Standard Quadratic Optimization Problems via Linear, Semidefinite and Copositive Programming / I. M. Bomze, E. Klerk // Journal of Global Optimization. – 2002. – № 24(2). – P. 163 – 185.

16. Borchers B. CSDP, a C library for semidefinite programming

[Електронний ресурс] / В. Borchers. – Електрон. дані (1 файл). – New Mexico Institute of Technology, 1999. – 10 р. – Режим доступу: euler.nmt.edu/~brian/csdpaper.pdf.

17. Borchers В. SDPLIB 1.2, A Library of Semidefinite Programming Test Problems [Електронний ресурс] / В. Borchers. – Електрон. дані (1 файл). – 8 р. – Режим доступу: euler.nmt.edu/~brian/sdplib/sdplib.pdf.

18. Bundfuss S. Copositive Matrices, Copositive Programming, and Applications [Електронний ресурс] / S. Bundfuss. – Електрон. дані (1 файл). – Darmstadt, 2009. – 150 р. – Режим доступу: <https://www3.mathematik.tu-darmstadt.de/index.php?id=483>.

19. Burer S. A Finite Branch-and-Bound Algorithm for Nonconvex Quadratic Programming via Semidefinite Relaxations [Електронний ресурс] / S. Burer, D. Vandebussche. – 2006. – 23 р. – Режим доступу: dollar.biz.uiowa.edu/~sburer/papers/017-qpbbs.pdf.

20. Burer S. A Nonlinear Programming Algorithm for Solving Semidefinite Programs via Low-rank Factorization [Електронний ресурс] / S. Burer, R. D. C. Monteiro. – Електрон. дані (1 файл). – 2001. – 30 р. – Режим доступу: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.3322&rep=rep1&type=ps.

21. Burer S. SDPLR 1.03-beta User's Guide [Електронний ресурс] / S. Burer. – Електрон. дані (1 файл). – 2009. – 5 р. – Режим доступу: dollar.biz.uiowa.edu/~sburer/files/SDPLR-1.03-beta-usrguide.pdf.

22. Cassioli A. Solving the Sensor Network Localization Problem using an Heuristic Multistage Approach [Електронний ресурс] / A. Cassioli. – Електрон. дані (1 файл). – Режим доступу: www.optimization-online.org/DB_FILE/2009/03/2267.pdf.

23. Chapter 4. Interior Point Methods [Електронний ресурс]. – Електрон. дані (1 файл). – Р. 59–80. – Режим доступу: <http://www.ti.inf.ethz.ch/ew/lehre/ApproxSDP09/notes/interior.pdf>.

24. Cobham M. The intrinsic computational difficulty of functions / M. Cobham // International Congress for Logic Methodology and Philosophy of

- Science. – Електрон. дані (1 файл). – Amsterdam: Y. Bar-Hillel, 1964. – P. 24–30.
25. COPSО: Constrained Optimization via PSO algorithm. Appendix A: Benchmark functions [Електронний ресурс] / А. Н. Aguirre, А. Е. М. Zavala, Е. V. Diharce, S. В. Rionda. – Електрон. дані (1 файл). – 2007. – P. 21–28. – Режим доступу: www.cimat.mx/reportes/enlinea/I-07-04.pdf.
26. Dattorro J. Convex Optimization and Euclidean Distance Geometry / J. Dattorro. – USA: Meboo, 2005. – P. 111–136, 269–391, 597–606.
27. Ding Y. On Equivalence of Semidefinite Relaxations for Quadratic Matrix Programming [Електронний ресурс] / Y. Ding, D. Ge, H. Wolkowicz. – Електрон. дані (1 файл). – 2010. – 23 р. – Режим доступу: www.optimization-online.org/DB_FILE/2010/04/2606.pdf.
28. Dur M. Copositive Programming – a Survey [Електронний ресурс] / M. Dur // Optimization Online. – Електрон. дані (1 файл). – 2009. – 17 р. – Режим доступу: www.springer.com/cda/content/document/cda_downloaddocument/9783642125973-c1.pdf.
29. Edmonds J. Paths, trees, and flowers [Електронний ресурс] / J. Edmonds. – Електрон. дані (1 файл). – 1965. – P. 449–467. – Режим доступу: www.disco.ethz.ch/lectures/fs12/seminar/paper/Tobias/2.pdf.
30. Epperly T.G.W. Global optimization test problem with solution [Електронний ресурс] / T.G.W. Epperly, R. E. Swaney. – Електрон. дані. – 1995. – 34 р. – Режим доступу: <http://citeseer.nj.nec.com/147308.html>. – Назва з екрану.
31. Exploiting Sparsity in Semidefinite Programming via Matrix Completion / Nakata K., Fujisawa K., Fukuda M., Kojima M., Murota K. – Tokyo: Tokyo Institute of Technology, 2001. – 24 р.
32. Feige U. Improved approximation algorithms for minimum-weight vertex separators [Електронний ресурс] / U. Feige, M. T. Hajiaghayi, J.R. Lee. – Електрон. дані (1 файл). – 2005. – 28 р. – Режим доступу: homes.cs.washington.edu/~jrl/papers/fhl-sicomp.pdf.
33. Fortin C. Computing the local minimizers of a large and sparse trust-region subproblem [Електронний ресурс] / C. Fortin. – Електрон. дані (1 файл). –

Montreal : McGill University, 2004. – 149 p. – Режим доступа: digitool.library.mcgill.ca/thesisfile85548.pdf.

34. Freund R. M. Interior Point Methods: Current Status and Future Direction [Электронный ресурс] / R. M. Freund, S. Mizuno. – Электрон. дані (1 файл). – 1998. – 29 p. – Режим доступа: web.mit.edu/rfreund/www/ipm.ps.

35. Freund R. M. Introduction to semidefinite programming [Электронный ресурс] / R. M. Freund. – Электрон. дані (1 файл). – Massachusetts Institute of Technology. – 2004. – 54 p. – Режим доступа: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/readings/MIT6_251JF09_SDP.pdf.

36. Furrer L. An Overview of Cutting Plane Methods for Semidefinite Programming [Электронный ресурс] / L. Furrer. – Электрон. дані (1 файл). – 2009. – 33 p. – Режим доступа: www.math.kth.se/optsys/grundutbildning/kurser/SF2827/furrer.pdf.

37. Gepp A. A Review of Semidefinite programming including Financial Applications / A. Gepp. – 2006. – 29 p.

38. Graph Laplacian Regularization for Large-Scale Semidefinite programming [Электронный ресурс] / Weinberger K. Q., Sha F., Zhu Q., Saul L. K. – Электрон. дані (1 файл). – 2007. – 8 p. – Режим доступа: <http://papers.nips.cc/paper/3060-graph-laplacian-regularization-for-large-scale-semidefinite-programming.pdf>.

39. Harrar D. L. Parallel solution of some large-scale eigenvalue problems arising in chemistry and physics [Электронный ресурс] / D. L. Harrar, M. R. Osborne // Applied Parallel Computing Large Scale Scientific and Industrial Problems. – Электрон. дані (1 файл). – 1998. – Vol. 1541. – P. 216–223. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.1365&rep=rep1&type=pdf>.

40. Helmberg C. A spectral bundle method for semidefinite programming / C. Helmberg, F. Rendl // SIAM J. Optim. – 2000. – vol.10. – P. 673–696.

41. Helmberg C. An interior point method for semidefinite programming /

C. Helmberg, F. Rendl, R. Vanderbei, H. Wolkowicz // *SIAM J. Optim.*, 1996. – vol. 6. – no. 2. – P. 342–361.

42. Helmberg C. Semidefinite Programming [Электронный ресурс] / C. Helmberg. – Электрон. дані (1 файл). – Berlin, 1999. – 20 p. – Режим доступу: <https://opus4.kobv.de/opus4-zib/files/438/SC-99-49.pdf>.

43. Helmberg C. Semidefinite Programming For Combinatorial Optimization [Электронный ресурс] / C. Helmberg. – Электрон. дані (1 файл). – Berlin, 2000. – 150 p. – Режим доступу: <https://opus4.kobv.de/opus4-zib/files/602/ZR-00-34.pdf>.

44. Hiriart-Urruty J.-B. A Variational Approach to Copositive Matrices [Электронный ресурс] / J.-B. Hiriart-Urruty, A. Seeger. – Электрон. дані (1 файл). – 43 p. – Режим доступу: www.math.univ-toulouse.fr/~jbhu/Copositivity_SIREV.pdf.

45. Jiawang Nie. Sum of Squares Method for Sensor Network Localization [Электронный ресурс] / N. Jiawang. – Электрон. дані (1 файл). – University of Minnesota – 2006. – Режим доступу: www.math.ucsd.edu/~njw/PUBLICPAPERS/sennetloc_cmopt.pdf.

46. Karger D. Approximate Graph Coloring by Semidefinite programming [Электронный ресурс] / D. Karger, R. Motwani, M. Sudan. – Электрон. дані (1 файл). – 18 p. – Режим доступу: <http://dl.acm.org/citation.cfm?id=274791>.

47. Kenneth V. P. Differential Evolution. A Practical Approach to Global Optimization / V. P. Kenneth, R. M. Storn, J. A. Lampinen. – Berlin Heidelberg: Springer-Verlag, 2005. – 542 p.

48. Kocvara M. PENNON – A Generalized Augmented Lagrangian Method for Semidefinite Programming / M. Kocvara, M. Stingl // *Appl. Optim.* – 2003. – № 82. – P.303–321.

49. Kocvara M. PENNON – A Code for Convex Nonlinear and Semidefinite Programming / M. Kocvara, M. Stingl // *Optimization Methods and Software.* – 2003. – №18(3). – P.317–333.

50. Kocvara M. PENSDP User's Guide (Version 2.2) [Электронный ресурс] / M. Kocvara, M. Stingl. – Электрон. дані (1 файл). – 2006. – 22 p. – Режим

доступу: www.penopt.com/doc/pensdp2_2.pdf.

51. Kosolap A. The upper and lower bounds for solutions of general quadratic optimization problems / Anatolii Kosolap, Anastasiia Peretiatko // Computer science & engineering: матеріали VI Міжнар. наук. конф. молодих вчених, 21-23 листопада 2013 р., м. Львів. – Львів: Видавництво Львівської політехніки, 2013. – С. 94–95.

52. Krishnan K. Properties of a Cutting Plane Method for Semidefinite Programming [Електронний ресурс] / K. Krishnan, J. E. Mitchell. – Електрон. дані (1 файл). – 2007. – 29 р. – Режим доступу: www4.ncsu.edu/~kksivara/publications/properties_revised.pdf.

53. Krislock N. Explicit Sensor Network Localization using Semidefinite Representations and Facial Reductions / N. Krislock, F. Wolkowicz. – 2010. – Vol. 20. – No. 5. – P. 2679–2708.

54. Kulis B. Convex Perturbations for Scalable Semidefinite Programming [Електронний ресурс] / B. Kulis, S. Sra, I. Dhillon // 12th International Conference on Artificial Intelligence and Statistics. – Електрон. дані (1 файл). – Florida: 2009. – 8 р. – Режим доступу: jmlr.csail.mit.edu/proceedings/papers/v5/kulis09a/kulis09a.pdf.

55. Lasserre J. B. Convergent LMI relaxations for nonconvex quadratic programs [Електронний ресурс] / J. B. Lasserre // 39th IEEE Conference on Decision and Control. – Електрон. дані (1 файл). – Sydney, 2000. – P. 5041–5046. – Режим доступу: www.mit.edu/~6.454/www_fall_2001/cmccaram/lasserre_1.pdf.

56. Laurent M. Semidefinite Optimization [Електронний ресурс] / Monique Laurent, Frank Vallentin. – Електрон. дані (1 файл). – Mastermath Spring, 2012. – 150 р. – Режим доступу: page.mi.fu-berlin.de/fmario/sdp/laurentv.pdf.

57. Laurent M. Semidefinite Programming and Integer Programming [Електронний ресурс] / M. Laurent, F. Rendl. – Електрон. дані (1 файл). – 2002. – 101 р. – Режим доступу: homepages.cwi.nl/~monique/files/chapter1.ps.

58. Lin C.-J. A predictor corrector method for semi-definite linear programming / C.-J. Lin, R. Saigal. – Michigan: The University of Michigan, 1995. – 23 р.

59. Lin C.-J. An Infeasible Start Predictor Corrector Method for Semi-definite Linear Programming / C.-J. Lin. – Ann Arbor: The University of Michigan, 1995. – 33 p.

60. Linear Matrix Inequalities in System and Control Theory [Электронный ресурс] / Boyd S., El Ghaoui L., Feron E., Balakrishnan V. – Электрон. дані (1 файл). – Philadelphia: SIAM, 1994. – 193 p. – Режим доступа: <https://web.stanford.edu/~boyd/lmibook/lmibook.pdf>.

61. Liu Z. An Infeasible Interior-Point Algorithm with Full Nesterov-Todd Step for Semidefinite Programming / Z. Liu. – Hohai: Hohai University, 2007. – 24 p.

62. Lopez A. A rank 1 SDP approach for the sensor network localization problem [Электронный ресурс] / Ania Lopez. – Электрон. дані (1 файл). – 2008. – 86 p. – Режим доступа: docserv.uni-duesseldorf.de/servlets/DerivateServlet/Derivate-10583/lopez.pdf.

63. Lovasz L. Semidefinite programs and combinatorial optimization [Электронный ресурс] / L. Lovasz. – Электрон. дані (1 файл). – Redmond: Microsoft Research. – 54 p. – Режим доступа: www.cs.elte.hu/~lovasz/semidef.ps.

64. Luo Z.-Q. Semidefinite Relaxation of Quadratic Optimization Problems [Электронный ресурс] / Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang // SP magazine, special issue on “Convex opt. for SP”. – Электрон. дані (1 файл). – 2010. – 14 p. – Режим доступа: www.se.cuhk.edu.hk/~mancho/papers/sdrapp-SPM.pdf.

65. MacQueen J. Some methods for classification and analysis of multivariate observations [Электронный ресурс] / J. MacQueen // In Proc. 5th Berkeley Symp. on Math. Statistics and Probability. – Электрон. дані (1 файл). – 1967. – P. 281–297. – Режим доступа: www.umiacs.umd.edu/~raghuram/ENEE731/Spectral/kMeans.pdf.

66. Man-Cho A. Theory of Semidefinite Programming for Sensor Network Localization [Электронный ресурс] / A. Man-Cho, Y. Ye. – Электрон. дані (1 файл). – ACM-SIAM Symposium on Discrete Algorithms, 2004. – 16 p. – Режим доступа: web.stanford.edu/~yyue/local-theory.pdf.

67. Mansouri H. A New Full-Newton Step $O(n)$ Infeasible Interior-Point

Algorithm for Semidefinite Optimization [Електронний ресурс] / Н. Mansouri, C. Roos. – Електрон. дані (1 файл). – Shahrekord University, 2008. – 29 р. – Режим доступу: www.optimization-online.org/DB_FILE/2008/07/2034.pdf.

68. Martello S. Knapsack problems: algorithms and computer implementation / S. Martello, P. Toth. – Chichester: John Wiley & SONS, 1990. – 296 р.

69. Mitchell J. E. Solving Semidefinite Programming Problems Using Branch-and-Bound [Електронний ресурс] / J. E. Mitchell. – Електрон. дані (1 файл). – Budapest: 2000. – 40 р. – Режим доступу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.9340&rep=rep1&type=pdf>.

70. Monteiro R. D. C. First- and Second-Order Methods for Semidefinite Programming / R. D. C. Monteiro // Math. Program. – 2003. – № 97. – 31 р.

71. Nesterov Y. E. Interior point polynomial algorithms in convex programming / Y. E. Nesterov, A. S. Nemirovskii // SIAM Studies in Applied Mathematics. – 1994. – Vol. 13. – SIAM: Philadelphia, USA. – 405 р.

72. Nie J. The Algebraic Degree of Semidefinite Programming [Електронний ресурс] / J. Nie, K. Ranestad, B. Sturmfels. – Електрон. дані (1 файл). – 23 р. – Режим доступу: www.math.ucsd.edu/~njw/PUBLICPAPERS/sdpalgdeg_mpa3.pdf.

73. Nocedal J. Numerical optimization [Електронний ресурс] / J. Nocedal, S.J. Wright. – Електрон. дані (1 файл). – Springer, 2006. – 685 р. – Режим доступу: www.bioinfo.org.cn/~wangchao/maa/Numerical_Optimization.pdf.

74. O'Donnell R. Advanced Approximation Algorithms. Lecture 14: Semidefinite Programming and Max-Cut [Електронний ресурс] / R. O'Donnell. – Електрон. дані (1 файл). – 2008. – 6 р. – Режим доступу: www.cs.cmu.edu/~anupamg/adv-approx/lecture14.pdf.

75. Overton M. Semidefinite Programming / M. Overton, H. Wolkowicz. // Mathematical Programming. – 1997. – № 77. – P. 105–109.

76. Pataki G. Cone-LP's and Semidefinite Programs: Geometry and a Simplex-type Method [Електронний ресурс] / G. Pataki. – Електрон. дані (1 файл). – University of Waterloo, 1999. – 13 р. – Режим доступу: www.unc.edu/~pataki/papers/ipcoV.pdf.

77. Peretiatko A. S. Using semidefinite simplex method for solving semidefinite problems / A. S. Peretiatko // Theoretical & Applied Science. – 2013. – № 12 (8). – P. 5–8.

78. Povh J. Copositive and Semidefinite Relaxations of the Quadratic Assignment Problem [Електронний ресурс] / J. Povh, F. Rendl. – Електрон. дані (1 файл). – 2006. – 15 р. – Режим доступу: www.math.uwaterloo.ca/~hwolkowi/henry/teaching/grad-studies/pprsgradseminarF08.d/copsdpqap08.pdf.

79. Prodi G. Eigenvalues of non-linear problems / G. Prodi (ed.). – Berlin Heidelberg: Springer-Verlag, 2010. – 238 p.

80. Qualizza A. Linear Programming Relaxations of Quadratic Programs [Електронний ресурс] / A. Qualizza, P. Belotti, F. Margot. – Електрон. дані (1 файл). – 19 р. – Режим доступу: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1260&context=tepper>.

81. Regularization Methods for Semidefinite Programming [Електронний ресурс] / Malick J., Povh J., Rendl F., Wiegele A. // SIAM Journal of Optimization. – Електрон. дані (1 файл). – 2009. – 26 р. – Режим доступу: www.optimization-online.org/DB_FILE/2007/10/1800.pdf.

82. Roumili H. Infeasible Interior Point Method for Semidefinite Programs / H. Roumili, A. Keraghel, A. Yassine // Applied Mathematical Sciences. – 2007. – Vol. 1. – No. 21. – P. 1009–1018.

83. Scalable Semidefinite Programming using Convex Perturbations / Kulis B., Sra S., Jegelka S., Dhillon I. – Austin: University of Texas, 2007. – 14 p.

84. SDPPACK, user's guide [Електронний ресурс] / Alizadeh F., Haeberly J. P., Nayakkankuppam M.V., Overton M.L. – Електронні дані (1 файл). – 1997. – 30 р. – Режим доступу: <http://cims.nyu.edu/~overton/software/sdppack/v0.8-beta/doc/userman.html>

85. Semidefinite Programming by Perceptron Learning [Електронний ресурс] / Graepel T., Herbrich R., Kharechko A., Shawe-Taylor J. – Електронні дані (1 файл). – 2005. – 8 р. – Режим доступу: papers.nips.cc/paper/2434-semi-definite-programming-by-perceptron-learning.pdf.

86. SFSDP: a Sparse Version of Full Semidefinite Programming Relaxation for Sensor Network Localization Problems [Електронний ресурс] / Kim S., Kojima M., Waki H., Yamashita M. – Електронні дані (1 файл). – 2009. – 19 р. – Режим доступу: www.is.titech.ac.jp/~kojima/articles/B-457.pdf.

87. Serali H. D. A Global Optimization RLT-based Approach for Solving the Hard Clustering Problem / H. D. Serali, D. Jitmitra // *Journal of Global Optimization*. – 2005. – № 32. – P. 281–306.

88. Sivaramakrishnan K. A conic interior point decomposition approach for large scale semidefinite programming [Електронний ресурс] / K. Sivaramakrishnan, G. Plaza, T. Terlaky. – Електронні дані (1 файл). – 2005. – 40 р. – Режим доступу: www4.ncsu.edu/~kksivara/publications/conic-decomposition-mathprog.pdf.

89. Sivaramakrishnan K. Linear programming approaches to semidefinite programming problems / K. Sivaramakrishnan. – New York: Rensselaer Polytechnic Institute, 2002. – 177 р.

90. So A. M. C. A semidefinite programming approach to the graph realization problem: theory, applications and extensions [Електронний ресурс] / A. M. C. So. – Електронні дані (1 файл). – Stanford University, 2007. – 116 р. – Режим доступу: www1.se.cuhk.edu.hk/~manchoso/papers/thesis.pdf.

91. SP: Software for Semidefinite Programming [Електронний ресурс]. – Електронні дані. – Режим доступу: <http://www.ee.ucla.edu/~vandenbe/sp.html>. – Назва з екрану.

92. Spaseloc: An Adaptive Subproblem Algorithm For Scalable Wireless Sensor Network Localization / Michael W. Carter, Holly H. Jin, M. A. Saunders, Y. Ye // *Siam J. Optim.* – 2006. – Vol. 17, № 4. – P. 1102–1128.

93. Stingl M. On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods [Електронний ресурс] / N. Stingl. – Електронні дані (1 файл). – 154 р. – Режим доступу: web.mat.bham.ac.uk/kocvara/pennon/michael_stingl_thesis.pdf.

94. Sturm J. S. Using SeDumi 1.02, A MatLab Toolbox for Optimization over Symmetric Cones [Електронний ресурс] / J. S. Sturm // *Optimization Methods and*

Software. – Електронні дані (1 файл). – 1999. – 30 р. – Режим доступу: sedumi.ie.lehigh.edu/wp-content/sedumi-downloads/usrguide.ps.

95. Todd M. J. On the Nesterov-Todd direction in semidefinite programming / M. J. Todd, K. C. Toh, R. H. Tutuncu // *SIAM Journal on Optimization*. – 1998. – № 8. – P. 769–796.

96. Todd M. J. *Semidefinite Optimization* / M. J. Todd. – Cambridge: Cambridge University Press, 2001. – 40 р.

97. Toh K. C. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming [Електронний ресурс] / K. C. Toh, R. H. Tutuncu, M. J. Todd. – Електронні дані (1 файл). – Draft, 2006. – 46 р. – Режим доступу: ww1.math.nus.edu.sg/events/AMC/papers-invited/Toh-Kim-Chuan.pdf.

98. Trafalis T. B. Computation of the regression kernel matrix using semidefinite programming / T. B. Trafalis, A. M. Malyscheff. – Norman: University of Oklahoma, 2006. – 14 р.

99. Vandenberghe L. *Semidefinite programming* / L. Vandenberghe, S. Boyd // *SIAM REVIEW*. – 1996. – Vol. 38, No. 1. – P. 49–95.

100. Wang M. Information Theoretical Clustering via Semidefinite Programming [Електронний ресурс] / Meihong Wang, Fei Sha // 14th International Conference on Artificial Intelligence and Statistics. – Електронні дані (1 файл). – USA, 2011. – vol. 15. – P. 761–769. – Режим доступу: jmlr.org/proceedings/papers/v15/wang11b/wang11b.pdf.

101. Wen Z. Alternating Direction Augmented Lagrangian Methods for Semidefinite Programming / Z. Wen, D. Goldfarb, W. Yin // *Math. Prog.* – 2010. – №2. – P. 203–230.

102. Wolkowicz H. *Semidefinite Programming. Research Report* / H. Wolkowicz. – Waterloo: University of Waterloo, 2002. – 17 р.

103. Wu S. P. SdpSol – a Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure [Електронний ресурс] / S. P. Wu, S. Boyd. – Електронні дані (1 файл). – 1996. – 17 р. – Режим доступу:

https://web.stanford.edu/~boyd/old_software/sdpsol/usrguide.ps.

104. Xia Q. Comparison Between an Infeasible Interior Point Algorithm and a Homogeneous Self Dual Algorithm for Semidefinite Programming / Q. Xia. – Socorro: New Mexico Institute of Mining and Technology, 2006. – 69 p.

105. Yamashita M. and others. A High-Performance Software Package for Semidefinite Programs: SDPA 7 [Электронный ресурс] / Yamashita M. and others. – Электронні дані (1 файл). – 2010. – 26 p. – Режим доступу: www.optimization-online.org/DB_FILE/2010/01/2531.pdf.

106. Yamashita M. Implementation and Evaluation of SDPA 6.0 / M. Yamashita, K. Fujisawa, M. Kojima. – Tokyo: Tokyo Institute of Technology, 2002. – 12 p.

107. Ye Y. Semidefinite programming / Y. Ye. – Stanford: Stanford University, 2003. – 161 p.

108. Yildirim E. A. An interior-point perspective on sensitivity analysis in linear programming and semidefinite programming / E. A. Yildirim. – Cornell: Cornell University, 2001. – 202 p.

109. Zhang Y. On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming / Y. Zhang // SIAM J. Optim. – 1998. – vol. 8. – P. 365–386.

110. Zhao X. Y. A Newton-CG Augmented Lagrangian Method for Semidefinite Programming [Электронный ресурс] / X. Y. Zhao, D. Sun, K. H. Toh. – Электронні дані (1 файл). – Hamilton, 2007. – 49 p. – Режим доступу: dspace.mit.edu/openaccess-disseminate/1721.1/58308.

111. Бабынин М.С. Барьерно-проективный метод для решения задач полуопределенного проектирования [Электронный ресурс] / М.С. Бабынин. – М.: 49-я научная конференция МФТИ, 2006. – Режим доступу: http://mipt.ru/upload/d07/f_9hup-arpetelja4w.pdf.

112. Гэри М. Вычислительные машины и труднорешаемые задачи [Электронный ресурс] / М. Гэри, Д. Джонсон. – Электрон. дані (1 файл). – М.: Мир, 1982. – Режим доступу: <http://cs.usu.edu.ru/hardness/data/books/NP->

Complectness.pdf.

113. Данциг Дж. Линейное программирование его обобщение и применение / Дж. Данциг; пер. с англ. Г.Н. Андрианова, Л.И. Горькова, А.А. Корбута, А.Н. Ляпунова; под ред. Н.Н. Воробьева. – М.: Прогресс, 1966. – 600 с.

114. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения / Дж. Деммель; пер. с англ. Х. Д. Икрамова. – М.: Мир, 2001. – 430 с.

115. Демьянов В.Ф. Недифференцируемая оптимизация / В.Ф. Демьянов, Л.В. Васильев. – М.: Наука, 1981. – 391 с.

116. Ермольев Ю.М. Методы решения невыпуклых негладких задач стохастической оптимизации / Ю.М. Ермольев, В.И. Норкин // Кибернетика и системный анализ. – 2003. – № 5. – С. 89–106.

117. Жадан В.Г. Прямой метод Ньютона для линейной задачи полуопределенного программирования / В. Г. Жадан // Труды института математики и механики. – 2008. – Том 4, № 2. – С. 67–80.

118. Жадан В.Г. Прямые методы внутренней точки с ускоренной сходимостью к допустимому множеству для задач полуопределенного программирования / В. Г. Жадан, М. С. Бабынин // Дискретная оптимизация и исследование операций: российская конференция, 7-14 сентября 2007 г., Владивосток. – Новосибирск: Издательство института математики, 2007. – С. 105.

119. Исковских В.И. Алгебраические поверхности / В. И. Исковских, И. Р. Шафаревич. – М.: Наука, 1989 – 138 с.

120. Коллатц Л. Задачи на собственные значения. С техническими приложениями / Л. Коллатц; пер. с нем. В. В. Никольского. – М.: Наука, 1968. – 504 с.

121. Кораблев Н.М. Кластеризация данных методом k-means с использованием искусственных иммунных систем / Н. М. Кораблев, А. А. Фомичев // Бионика интеллекта. – 2011. – № 3 (77). – С. 102–106.

122. Косолап А. И. Верхние и нижние оценки решений в общих задачах квадратичной оптимизации / А. И. Косолап, А. С. Перетяцько // Вісник

Харківського національного університету імені В.Н. Каразіна. Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». – 2013. – Вип. 23, № 1089. – С. 96–102.

123. Косолап А.И. Выпуклый анализ и многоэкстремальные задачи / А.И. Косолап. – Днепропетровск: Издательство ДНУ, 2007 – 280 с.

124. Косолап А.И. Методы глобальной оптимизации / А.И. Косолап. – Днепропетровск: Наука и образование, 2013. – 316 с.

125. Косолап А.И. Обобщение симплекс-метода для решения задач полуопределенной оптимизации / А.И. Косолап // Математичне та комп'ютерне моделювання. – 2010. – № 3. – С. 99–106.

126. Косолап А. И. Поиск максимально плотной упаковки сфер в пространстве / А. И. Косолап, А. С. Перетяцько // Хімія та сучасні технології: тези доповідей VI Міжнар. наук.-техн. конф. студентів, аспірантів та молодих вчених, 24-26 квітня 2013 р., м. Дніпропетровськ. – Дніпропетровськ: Український державний хіміко-технологічний університет, 2013. – т. 3. – С. 121–122.

127. Косолап А. И. Полуопределенная оптимизация в задаче расположения датчиков в сети / Косолап А. И., Перетяцько А. С. // Математичні машини і системи. – 2014. – № 2. – С. 105–112.

128. Косолап А. И. Полуопределенное программирование для решения задач комбинаторной оптимизации / А. И. Косолап, А. С. Перетяцько // Вісник Запорізького національного університету. Серія «Прикладна математика. Інформатика». – 2013. – № 2. – С. 50–55.

129. Косолап А. И. Полуопределенный симплекс-метод / А. И. Косолап, А. С. Перетяцько // Вісник Черкаського університету. Серія «Прикладна математика. Інформатика». – 2013. – № 18 (271). – С. 50–56.

130. Косолап А. И. Применение задачи полуопределенной оптимизации в задаче локализации датчиков в сети / А. И. Косолап, А. С. Перетяцько // Математичне та імітаційне моделювання систем: тези доповідей VII Міжнар. наук.-практ. конф., 25-28 червня 2012 р., м. Чернігів-Жукин. – Чернігів:

Чернігівський державний технологічний університет, 2012. – С. 300–303.

131. Косолап А. И. Сопряженные направления в задачах на собственные значения симметричных матриц / А. И. Косолап, А. С. Перетяцько // Проблеми обчислювальної механіки і міцності конструкцій: зб. наук. праць. – Дніпропетровськ: Ліра, 2013. – Вип. 21. – С. 114–122.

132. Косолап А. И. Численная эффективность методов полуопределенной оптимизации / А. И. Косолап, А. С. Перетяцько // Проблемы управления и информатики. – 2014. – № 2. – С. 56–64.

133. Косолап А.І. Використання напіввизначеної оптимізації для розв'язування задачі розташування датчиків у мережі / А.І. Косолап, А.С. Перетяцько // Современные проблемы математики и ее приложения в естественных науках и информационных технологиях: VIII Міжнародна наукова конференція для молодих вчених, 27-28 квітня 2013 р.: тези доповідей / Харківський національний університет ім. В. Н. Каразіна. – Харків: ХНУ, 2013. – С. 84–85.

134. Косолап А.І. Використання симплекс-методу для розв'язання задач напіввизначеного програмування / А.І. Косолап, А.С. Перетяцько // Математичне та програмне забезпечення інтелектуальних систем MPZIS-2009: VII Міжнародна науково-практична конференція, 25-27 листопада 2009 р.: тези доповідей / Дніпропетровський національний університет імені Олеся Гончара. – Дніпропетровськ, 2009. – С. 134–135.

135. Косолап А. І. Збігання узагальненого симплекс-методу для напіввизначеної оптимізації / А. І. Косолап, А. С. Перетяцько // Інформатика та системні науки: матеріали III Всеукр. наук.-практ. конф., 1-3 березня 2012 р., м. Полтава. – Полтава: ПУЕТ, 2012. – С. 153–156.

136. Косолап А.І. Напіввизначена оптимізація для моделювання складних систем / А.І. Косолап, А.С. Перетяцько // Математичне та імітаційне моделювання систем МОДС 2011: VI Науково-практична конференція з міжнародною участю, 27-30 червня 2011 р.: тези доповідей. – Чернігів, 2011. – С. 370–374.

137. Косолап А. І. Напіввизначена оптимізація для моделювання складних систем / А. І. Косолап, А. С. Перетяцько // Математичні машини і системи. – 2012. – № 1. – С. 174–179.

138. Косолап А.І. Напіввизначена оптимізація у задачі розташування датчиків у мережі з урахуванням відхилень / А.І. Косолап, А.С. Перетяцько // Математичне та імітаційне моделювання систем МОДС 2013: VIII міжнародна науково-практична конференція – Чернігів-Жукин, 24-28 червня 2013 року. – С. 154–155.

139. Косолап А.І. Напіввизначена релаксація в моделюванні складних систем / А.І. Косолап, А.С. Перетяцько // Контроль і управління в складних системах КУСС-2012: XI Міжнародна конференція, 9-11 жовтня 2012 р.: тези доповідей / Вінницький національний технічний університет. – Вінниця: ВНТУ, 2012. – С. 21.

140. Косолап А.І. Напіввизначена релаксація в моделюванні складних систем / А.І. Косолап, А.С. Перетяцько // Вісник Вінницького Політехнічного інституту. – 2013. – № 2(107). – С. 92–95.

141. Косолап А.І. Симлекс-метод для розв'язування задач напіввизначеного програмування / А.І. Косолап // Вісник Донецького Національного університету. – Донецьк: 2010. – С. 437–442.

142. Косолап А. І. Точність напіввизначеної релаксації для задач максимізації норми вектора [Електронний ресурс] / А. І. Косолап, А. С. Перетяцько // Інформатика та системні науки: матеріали VI Всеукр. наук.-практ. конф. за міжнар. участю, 19-21 березня 2015 р., м. Полтава. – Електрон. дані (1 файл). – Полтава: ПУЕТ, 2015. – [3 с.]. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/2489>.

143. Ланкастер П. Теория матриц / П. Ланкастер. – М.: Наука, 1982. – 272 с.

144. Лэсдон Л.С. Оптимизация больших систем / Л.С. Лэсдон. – М.: Наука, 1975. – 432 с.

145. Ляшко С.И. Моделирование и оптимизация подземного массопереноса / С.И. Ляшко, Д.А. Ключин, А.С. Тригуб. – К.: Наукова думка,

1998. – 238 с.

146. Мандель И.Д. Кластерный анализ / И.Д. Мандель. – М.: Финансы и статистика. – 1988. – 176 с.

147. Михалевич В.С. Вычислительные методы выбора оптимальных решений / В.С. Михалевич, Н.З. Шор. – К.: Наукова думка, 1977. – 178 с.

148. Михалевич В.С. Методы невыпуклой оптимизации / В.С. Михалевич, А.М. Гупал, В.И. Норкин. – М.: Наука, 1987. – 280 с.

149. Норкин В.И. Устойчивость стохастических оптимизационных моделей и статистические методы стохастического программирования / В.И. Норкин. – Киев: Институт кибернетики им. В.М.Глушкова НАНУ, 1989. – 24 с.

150. Ортега Дж. Введение в численные методы решения дифференциальных уравнений / Дж. Ортега, У. Пул; пер. с англ. Н. Б. Конюховой; под ред. А. А. Абрамова. – М.: Наука, 1986. – 288 с.

151. Парлетт Б. Симметричная проблема собственных значений. Численные методы / Б. Парлетт; пер. с англ. Х.Д. Икрамова и Ю.А. Кузнецова. – М.: Мир, 1983. – 384 с.

152. Перетяцько А. С. Використання напіввизначеної оптимізації для розв'язку задачі пошуку максимального розрізу графа / А. С. Перетяцько, А. І. Косолап // Dynamical system modelling and stability investigation: abstracts of conference reports of XV International Conference, May 25-27 2011, Kyiv. – К.: Taras Shevchenko National University of Kyiv, 2011. – С. 376.

153. Перетяцько А. С. Использование полуопределенной релаксации для решения задачи кластеризации данных / А. С. Перетяцько // Актуальные проблемы современной науки: сборник статей Междунар. науч.-практ. конф., 13-14 декабря 2013 г. – Уфа: РИЦ БашГУ, 2013. – т. 2. – С. 307–310.

154. Перетяцько А. С. Напіввизначена оптимізація для розв'язку загальних квадратичних задач / А. С. Перетяцько // Інформатика та системні науки: матеріали V Всеукр. наук.-практ. конф. за міжнар. участю, 13-15 березня 2014 р., м. Полтава. – Полтава: ПУЕТ, 2014. – С. 240–243.

155. Перетяцько А. С. Обобщенный симплекс-метод для решения задач полуопределенной оптимизации / А. С. Перетяцько, А. И. Косолап // Обчислювальна та прикладна математика: матеріали IV Міжнар. конф. ім. академіка І. І. Ляшка, 8-10 вересня 2011 р., м. Київ. – К.: Київський національний університет ім. Тараса Шевченка, 2011. – С. 110.

156. Перетяцько А. С. Эффективность полуопределенной релаксации для решения общих квадратичных задач / А. С. Перетяцько // Обчислювальна та прикладна математика: матеріали VII Міжнар. наук. конф. ім. академіка І. І. Ляшка, 9-10 жовтня 2014 р., м. Київ. – К.: Київський національний університет ім. Тараса Шевченка, 2014. – С. 82.

157. Попов Л.Д. Введение в теорию, методы и экономические приложения задач о дополнителности: учебное пособие / Л.Д. Попов. – Екатеринбург: Издательство Уральского университета, 2001. – С. 10-12.

158. Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи / Б.Н. Пшеничный. – М.: Наука, 1986. – 234 с.

159. Пшеничный Б.Н. Численные методы в экстремальных задачах / Б.Н. Пшеничный, Ю.Н. Данилин. – М.: Наука, 1975. – 320 с.

160. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации / И.В. Сергиенко. – К.: Наукова думка, 1985. – 384 с.

161. Сергієнко І.В. Математичне моделювання в комп'ютерній томографії з використанням інтерфлотації функцій: монографія / І.В. Сергієнко, О.М. Литвин, Ю.І. Першина. – Харків: Українська інженерно-педагогічна академія, 2008. – 161 с.

162. Сокурєнко В.М. Числове дослідження стохастичних методів безперервної глобальної оптимізації / В.М. Сокурєнко, В.С. Неділюк // Наукові вісті НТУУ «КПШ». – 2012. – №1. – С. 81–87.

163. Стоян Ю.Г. Теория и методы евклидовой комбинаторной оптимизации / Ю.Г. Стоян, О.А. Емец. – К.: МКВО, 1993. – 188с.

164. Ту Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес; пер. с англ. И.Б. Гуревича; под. ред. Ю.И. Журавлева. – М.: Мир, 1978. – 413 с.

165. Уилкинсон Дж.Х. Алгебраическая проблема собственных значений / Дж.Х. Уилкинсон. – М.: Наука, 1970. – 564 с.
166. Хант Э. Искусственный интеллект / Э. Хант; пер с англ. Д.А. Белова и Ю.И. Крюкова; под. ред. В.Л. Стефанюка. – М.: Мир, 1978. – 560 с.
167. Хачиян Л.Г. Полиномиальный алгоритм в линейном программировании / Л.Г. Хачиян // Доклады АН СССР. – 1979. – т. 244. – С.1093-1096.
168. Хорн Р. Матричный анализ / Р. Хорн, Ч. Джонсон. – М.: Мир, 1989. – 656 с.
169. Шор Н.З. Квадратичные оптимизационные задачи / Н.З. Шор // Техническая кибернетика. – 1987. – № 1. – С. 128–139.
170. Шор Н.З. Квадратичные экстремальные задачи и недифференцируемая оптимизация / Н.З. Шор, С.И. Стеценко. – К.: Наукова думка, 1989. – 208 с.
171. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения / Н.З. Шор. – К.: Наукова думка, 1979. – 200 с.

ДОДАТКИ

Додаток А

Приклади, для яких метод внутрішньої точки не може знайти розв'язок

В наступних прикладах розв'язується пряма задача напіввизначеної оптимізації (1.1). Початкові дані представлені у вигляді таблиці: матриця C , нижче – матриці $A_i | b_i$. Розв'язок, який отримано напіввизначеним симплекс-методом, виводиться рядком нижче: матриця X та значення цільової функції $C \bullet X$. Прямо-двоїсті методи внутрішньої точки знаходять недопустимий розв'язок цих задач.

Приклад 1

Семивимірна задачі напіввизначеної оптимізації з вісьмома обмеженнями (дані представлені в табл. А.1).

Таблиця А.1 – Вхідні дані

$C =$	0	3,25	-0,5	-1	-1,5	-1	-0,5	
	3,25	-0,5	0	0	0	0	0	
	-0,5	0	0	0	0	0	0	
	-1	0	0	0	0	0	0	
	-1,5	0	0	0	0	0	0	
	-1	0	0	0	0	0	0	
	-0,5	0	0	0	0	0	0	
$A_1 =$	0	0,5	1	4	0,5	1,5	2,5	16
	0,5	0	0	0	0	0	0	
	1	0	0	0	0	0	0	
	4	0	0	0	0	0	0	
	0,5	0	0	0	0	0	0	
	1,5	0	0	0	0	0	0	
	2,5	0	0	0	0	0	0	
$A_2 =$	0	-0,05	0,25	1	2,5	-2,5	1,5	3
	-0,05	0	0	0	0	0	0	
	0,25	0	0	0	0	0	0	
	1	0	0	0	0	0	0	
	2,5	0	0	0	0	0	0	
	-2,5	0	0	0	0	0	0	
	1,5	0	0	0	0	0	0	

Приклад 2

Розглянемо приклад семивимірної задачі напіввизначеної оптимізації з вісьмома обмеженнями (дані представлені в табл. А.2).

Таблиця А.2 – Вхідні дані

$C =$	0	-5,25	-3,75	-1,75	-1,25	-0,75	-5	
	-5,25	-0,5	0	0	0	0	0	
	-3,75	0	-0,5	0	0	0	0	
	-1,75	0	0	-0,5	0	0	0	
	-1,25	0	0	0	-0,5	0	0	
	-0,75	0	0	0	0	-0,5	0	
	-5	0	0	0	0	0	0	
$A_1 =$	0	5	0	5	0	0	0,5	20
	5	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0,5	0	0	0	0	0	0	
$A_2 =$	0	-0,5	0	0	0	0	0	0
	-0,5	1	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
$A_3 =$	0	0	-0,5	0	0	0	0	0
	0	0	0	0	0	0	0	
	-0,5	0	1	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
$A_4 =$	0	0	0	-0,5	0	0	0	0
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	-0,5	0	0	1	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	

Продовження таблиці А.2

$A_5 =$	0	0	0	0	-0,5	0	0	0
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	-0,5	0	0	0	1	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
$A_6 =$	0	0	0	0	0	-0,5	0	0
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	-0,5	0	0	0	0	1	0	
	0	0	0	0	0	0	0	
$A_7 =$	0	0	0	0	0	0	-10	0
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	-10	0	0	0	0	0	0	1
$A_8 =$	1		0	0	0	0	0	1
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	

Результати розв'язування представлені в табл. А.3.

Таблиця А.3 – Матриця розв'язку

$X =$	0,999999	8,34E-05	0,99692	0,000524	0,998045	0,997081	19,99393
	8,34E-05	8,34E-05	0,00011	-0,00011	0,000119	-0,00013	0,002296
	0,99692	0,00011	0,99692	8,97E-05	0,994214	0,994568	19,92056
	0,000524	-0,00011	8,97E-05	0,000524	0,000293	0,000545	0,008636
	0,998045	0,000119	0,994214	0,000293	0,998045	0,995218	19,95992
	0,997081	-0,00013	0,994568	0,000545	0,995218	0,997081	19,93285
	19,99393	0,002296	19,92056	0,008636	19,95992	19,93285	399,8786

Значення цільової функції, яке було знайдено напіввизначеним симплекс-методом, дорівнює $C \bullet X = -212,906$. Усі обмеження виконані.

Приклад 3

Розглянемо приклад чотиривимірної задачі напіввизначеної оптимізації з шістьма обмеженнями (дані представлені в табл. А.4).

Таблиця А.4 – Вхідні дані

$C =$	1000	0	0	0	
	0	-1	-0,5	-0,5	
	0	-0,5	-2	0	
	0	-0,5	0	-1	
$A_1 =$	0	0	0	0	25
	0	1	0	0	
	0	0	1	0	
	0	0	0	1	
$A_2 =$	0	4	7	3,5	56
	4	0	0	0	
	7	0	0	0	
	3,5	0	0	0	
$A_3 =$	0	-5	0	0	0
	-5	1	0	0	
	0	0	0	0	
	0	0	0	0	
$A_4 =$	0	0	-5	0	0
	0	0	0	0	
	-5	0	1	0	
	0	0	0	0	
$A_5 =$	0	0	0	-5	0
	0	0	0	0	
	0	0	0	0	
	-5	0	0	1	
$A_6 =$	1	0	0	0	1
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

Результати розв'язування представлені в табл. А.5.

Таблиця А.5 – Матриця розв'язку

$X =$	1	1,46E-07	4	6,99E-07
	1,46E-07	1,46E-06	0,002618	2,59E-06
	4	0,002618	24,99999	0,006262
	6,99E-07	2,59E-06	0,006262	6,99E-06

Значення цільової функції, яке було знайдено напіввизначеним симплекс-методом, дорівнює $C \bullet X = 949,997$. Усі обмеження виконані.

Приклад 4

Розглянемо приклад десятивимірної задачі напіввизначеної оптимізації з чотирнадцятьма обмеженнями (дані представлені в табл. А.6).

Таблиця А.6 – Вхідні дані

$C =$	0	3	8	0	0	-4,5	5	5	-7,5	0	
	3	0	0	0	0	0	0	0	0	0	
	8	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	-4,5	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	0	0	0	
	-7,5	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
$A_1 =$	0	0	0	0	0	-0,0175	0,01	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0,5
	0	0	0	0	0	0	0	0	0	0	
	-0,0175	0	0	0	0	0	0	0	0	0	
	0,01	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
0	0	0	0,5	0	0	0	0	0	0		
$A_2 =$	0	0	0	0	0	0	0	0,01	-0,0075	0	0
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0,5
	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	
	0,01	0	0	0	0	0	0	0	0	0	
	-0,0075	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0,5	0	0	0	0	0	

Закінчення таблиці А.6

$A_{11} =$	0	0	0	0,5	0	-0,5	0,5	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0,5	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	-0,5	0	0	0	0	0	0	0	0	0	0
	0,5	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
$A_{12} =$	0	0	0	0	0,5	0	0	0,5	-0,5	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0,5	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0,5	0	0	0	0	0	0	0	0	0	0
	-0,5	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
$A_{13} =$	0	0,015	0,005	0	0	0	0	0	0	0	0
	0,015	0	0	0	0	0	0	0	0	0	0
	0,005	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	-0,5	0
	0	0	0	0	0	0	0	0	0	-0,5	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	-0,5	-0,5	0	0	0	0	0	0
$A_{14} =$	1	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0

Результати розв'язування представлені в табл. А.7.

Таблиця А.7 – Матриця розв'язку

$X =$	1,000	0,786	-0,088	0,516	0,182	0,889	0,372	0,737	0,918	0,022
	0,786	235,742	-0,027	0,199	0,174	0,807	0,427	0,782	0,651	0,016
	-0,088	-0,027	0,914	0,750	-0,255	0,446	0,316	0,158	0,270	0,004
	0,516	0,199	0,750	51,633	-0,297	0,755	0,510	0,390	1,178	0,024
	0,182	0,174	-0,255	-0,297	0,720	0,223	0,087	0,303	0,282	-0,001
	0,889	0,807	0,446	0,755	0,223	88,857	0,624	2,747	1,103	0,021
	0,372	0,427	0,316	0,510	0,087	0,624	111,672	0,499	111,836	0,010
	0,737	0,782	0,158	0,390	0,303	2,747	0,499	73,679	71,852	0,016
	0,918	0,651	0,270	1,178	0,282	1,103	111,836	71,852	183,688	0,024
	0,022	0,016	0,004	0,024	-0,001	0,021	0,010	0,016	0,024	0,001

Значення цільової функції, яке було знайдено напіввизначеним симплекс-методом, дорівнює $C \bullet X = -7,374$. Усі обмеження виконані.

Приклад 5

Розглянемо приклад шестивимірної задачі напіввизначеної оптимізації з сімома обмеженнями (дані представлені в табл. А.8).

Таблиця А.8 – Вхідні дані

$C =$	-40792,1	18,64662	0	0	0	0	
	18,64662	0	0	0	0	0,417845	
	0	0	0	0	0	0	
	0	0	0	5,357855	0	0	
	0	0	0	0	0	0	
	0	0,417845	0	0	0	0	
$A_1 =$	0	0	0	0	0	0	6,665593
	0	0	0	0	0,000313	0	
	0	0	0	0	0	0,002843	
	0	0	0	0	0	-0,0011	
	0	0,000313	0	0	0	0	
	0	0	0,002843	-0,0011	0	0	
$A_2 =$	0	0	0	0	0	0	-10,699
	0	0	0	-0,00063	0	0	
	0	0	0	0	0	0	
	0	-0,00063	0	0	-0,00095	-0,00235	
	0	0	0	-0,00095	0	0	
	0	0	0	-0,00235	0	0	

Продовження таблиці А.8

$A_3 =$	0	-90	0	0	0	0	-7956
	-90	1	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
$A_4 =$	0	0	-39	0	0	0	-1485
	0	0	0	0	0	0	
	-39	0	1	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
$A_5 =$	0	0	0	-37,5	0	0	-1350
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	-37,5	0	0	1	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
$A_6 =$	0	0	0	0	-36	0	-1215
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	-36	0	0	0	1	0	
	0	0	0	0	0	0	
$A_7 =$	1	0	0	0	0	0	1
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	
	0	0	0	0	0	0	

Результати розв'язування представлені в табл. А.9.

Таблиця А.9 – Матриця розв'язку

$X =$	1,000	78,828	34,927	30,418	42,384	35,801
	78,828	6232,975	2752,481	2390,613	3316,878	2755,109
	34,927	2752,481	1239,277	1060,997	1476,421	1238,205
	30,418	2390,613	1060,997	931,313	1295,090	1111,695
	42,384	3316,878	1476,421	1295,090	1836,637	1604,919
	35,801	2755,109	1238,205	1111,695	1604,919	1535,823

Продовження таблиці А.10

$A_4 =$	0	50	0	0	416,6663	0	0	0	0	83333,33
	50	0	0	0	0	0	-0,5	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	416,6663	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	-0,5	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
$A_5 =$	0	0	0	0	-625	625	0	0	0	0
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0,5	0	0	-0,5	0	
	0	0	0	0	0	0	0	0	0	
	-625	0	0,5	0	0	0	0	0	0	
	625	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	-0,5	0	0	0	0	0	0	
$A_6 =$	0	0	0	0	0	-1250	0	0	0	-1250000
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0,5	0	0	-0,5	
	0	0	0	0	0	0	0	0	0	
	-1250	0	0	0,5	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
$A_7 =$	1	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	

Результати розв'язування представлені в табл. А.11.

Значення цільової функції, яке було знайдено напіввизначеним симплекс-методом, дорівнює $C \bullet X = -2,816$. Усі обмеження виконані.

Таблиця А.11 – Матриця розв'язку

$X =$	1,00	6,8	-4,8	-4,8	99,0	-37,7	10,7	12,8	50,8
	6,8	297,0	33,3	33,3	2127,6	210,0	-123,2	-61,7	-265,7
	-4,8	33,3	170966,8	40,9	170832,4	306,5	-104,2	-102,0	-408,4
	-4,8	33,3	40,9	1345048,3	-93,3	-1344700,7	-104,2	-102,0	-408,4
	99,0	2127,6	170832,4	-93,3	189133,7	-1021,4	-75,6	406,3	1477,1
	-37,7	210,0	306,5	-1344700,7	-1021,4	1347306,5	-774,5	-766,3	-3065,5
	10,7	-123,2	-104,2	-104,2	-75,6	-774,5	270,8	256,6	1029,9
	12,8	-61,7	-102,0	-102,0	406,3	-766,3	256,6	255,6	1022,2
	50,8	-265,7	-408,4	-408,4	1477,1	-3065,5	1029,9	1022,2	4088,1

Додаток Б

Вхідні дані для задач

Таблиця Б.1 – Вхідні дані

$\tilde{Q} =$	0	0	0	0	0	0	0	
	0	1	0	0	0	0	0	
	0	0	1	0	0	0	0	
	0	0	0	1	0	0	0	
	0	0	0	0	1	0	0	
	0	0	0	0	0	1	0	
	0	0	0	0	0	0	0	1
$\tilde{A}_1 =$	1	-2	0	0	-2	-0,5	-0,5	0
	-2	8	-1	3	4	2	1	
	0	-1	-4	-2	1	2	-3	
	0	3	-2	-1	2	-1	1	
	-2	4	1	2	8	1	-1	
	-0,5	2	2	-1	1	-3	2	
	-0,5	1	-3	1	-1	2	-8	
$\tilde{A}_2 =$	10	-2	0	-3	1	-0,5	-2	0
	-2	-10	-1	1	4	2	1	
	0	-1	-6	-2	1	3	-3	
	-3	1	-2	6	2	-1	1	
	1	4	1	2	-6	1	-1	
	-0,5	2	3	-1	1	7	2	
	-2	1	-3	1	-1	2	8	
$\tilde{A}_3 =$	-20	-2	0	0	-2	-0,5	0,5	0
	-2	-12	-1	3	4	2	1	
	0	-1	-10	-2	1	2	-3	
	0	3	-2	2	2	-1	1	
	-2	4	1	2	8	1	2	
	-0,5	2	2	-1	1	-10	2	
	0,5	1	-3	1	2	2	8	

Додаток В

ЗАТВЕРДЖУЮ



Проректор з науково-педагогічної
роботи ДВНЗ «Український державний
хіміко-технологічний університет»
д.т.н., проф. В.І. Голеус

20 жовтня 2014р.

АКТ

**про впровадження результатів дисертаційної роботи
Перетяцько Анастасії Сергіївни
на тему "Напіввизначена оптимізація для розв'язування загальних
квадратичних задач"
у навчальний процес**

Цим актом засвідчується, що за результатами дисертаційної роботи Перетяцько А.С. «Напіввизначена оптимізація для розв'язування загальних квадратичних задач» у навчальний процес підготовки бакалаврів, спеціалістів та магістрів за напрямком 6.050102 – «Комп'ютерна інженерія» впроваджено наступне:

- теоретичні результати дисертаційної роботи та розроблені програми «Напіввизначений симплекс-метод» та «Прямо-двоїстий метод внутрішньої точки» – у курс лекцій «Аналіз, синтез і моделювання систем»;
- при підготовці дипломних робіт та проектів використовуються математичні моделі локалізації датчиків у мережі та кластеризації даних.

Результати впровадження розглянуті на засіданні науково-методичної ради Державного вищого навчального закладу «Український державний хіміко-технологічний університет».

Згідно з оригіналом

Вчений секретар спецради Д 64.052.02

Л.В. Колесник

Начальник
навчально-наукового центру,
к.т.н., доц.

Р.В. Смотров

Зав. каф. спеціалізованих
комп'ютерних систем, д.т.н., с.н.с.

Л.Д. Чумаков