НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ КІБЕРНЕТИКИ ім. В. М. ГЛУШКОВА
Вищий навчальний заклад Укоопспілки
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

Г. П. Донець
Л. М. Колєчкіна

ЕКСТРЕМАЛЬНІ ЗАДАЧІ НА КОМБІНАТОРНИХ КОНФІГУРАЦІЯХ

Монографія

ПОЛТАВА
РВВ ПУЕТ
2011

УДК 519.85
ББК 22.18
Д67

Рекомендовано до видання, розміщення в електронній бібліотеці та використання в навчальному процесі вченою радою університету, протокол № 9 від 16 листопада 2010 р.

Автори:
Г. П. Донець, Л. М. Колєчкіна

Рецензенти: ***Ф. І. Андон,*** д.ф.-м.н., академік НАН України, завідувач відділу економічної кібернетики Інституту кібернетики ім. В. М. Глушкова; ***С. В. Яковлев,*** д.ф.-м.н., професор, завідувач кафедри вищої математики та прикладної інформатики Харківського інституту управління, заслужений діяч науки і техніки України.

Донець Г. П.
Д67 Екстремальні задачі на комбінаторних конфігураціях : монографія / Г. П. Донець, Л. М. Колєчкіна. – Полтава : РВВ ПУЕТ, 2011. – 309 с.

ISBN 978-966-184-133-7

Монографія присвячена розробці та обґрунтуванню математичних моделей і обчислювальних методів розв’язання екстремальних задач дискретної оптимізації на комбінаторних конфігураціях. Особлива увага приділена дослідженню структурних властивостей допустимої області, побудові графів многогранників комбінаторних конфігурацій, що є областями допустимих розв’язків. Побудовано та обґрунтовано нові підходи й ефективні методи розв’язання екстремальних задач на комбінаторних конфігураціях при умові додаткових обмежень, проведено аналіз обчислювального експерименту.

Монографія розрахована на широке коло читачів-математиків, фахівців з математичного моделювання, кібернетики, теорії та практики векторної та дискретної оптимізації, а також студентів і аспірантів відповідних спеціальностей.

УДК 519.85
ББК 22.18

ISBN 978-966-184-133-7

ЗМІСТ

ВСТУП

Останнім часом з'явилася велика кількість праць, присвячених методам розв'язування екстремальних задач на комбінаторних конфігураціях [1–5, 11, 13, 53, 55–60, 86, 88–90, 94–99, 110, 112, 114–118, 120–127, 130, 137, 145–147, 171, 173, 176, 177, 180, 188, 193–196, 200, 202–205, 210–220, 223, 225, 226–228, 272–277, 283], в яких пропонуються нові підходи та удосконалюються існуючі методи, досліджуються їх ефективність. Даний факт пов'язаний з тим, що екстремальні задачі дискретної оптимізації є моделями різних прикладних задач проектування, планування, розміщення, класифікації і управління. Слід зазначити, що в науковій літературі широко вживається термін «задача комбінаторної оптимізації» (ЗКО), під якою розуміють проблему пошуку екстремумів заданої цільової функції на комбінаторному просторі. Під комбінаторним простором тут розуміється сукупність комбінаторних об'єктів певного типу, утворених із елементів заданої скінченної множини. Водночас, в більшості робіт з даного напряму, поняття «комбінаторний об'єкт» не формалізується, а, як правило, найчастіше називаються такі множини як сполучення, перестановки та розміщення. Але для більш строгої формалізації поняття комбінаторного об'єкту може бути здійснене на основі поняття комбінаторної конфігурації.

Отже в подальшому слід розуміти, що комбінаторні конфігурації та комбінаторні множини вживаються в одному й тому ж розумінні, в залежності від специфіки та властивостей задачі, хоча поняття комбінаторної конфігурації є більш об'ємним з точки зору комбінаторики.

Прикладами екстремальних задач є задача комівояжера, транспортна задача, задача розміщення об'єктів, трасування друкованих плат, що виникають при проектуванні обчислювальної апаратури, в автоматизованих системах управління тощо. Моделями таких задач є велике число прикладних задач, що виникають на виробництві [31, 33, 38–40, 52, 53, 92–94, 129, 146, 150–152, 160–166, 239, 268].

Необхідність розв'язання саме цих задач стимулювало розвиток теорії дискретної, зокрема комбінаторної оптимізації. Комбінаторна оптимізація – область математики, предметом якої є дослідження і розв'язання екстремальних задач на скінченних множинах комбінаторного характеру.

Основна увага в комбінаторній оптимізації приділяється визначенню обчислювальної складності цих задач, розробці методів і алгоритмів їх розв'язання на основі властивостей комбінаторних множин.

Існує безліч різноманітних методів розв'язування екстремальних задач, як точних так і наближених. Систематичне вивчення властивостей комбінаторних множин та їх дослідження описані в багатьох роботах [9, 14, 18–20, 36, 43–53, 58–60, 95, 136, 141–148, 166–169, 238–246, 253–256]. Підвищений інтерес до таких задач обумовлений дослідженнями останніх років в області комп'ютерних технологій при створенні сучасних алгоритмів і програм для розв'язування оптимізаційних задач. Слід зазначити, що задачі комбінаторної оптимізації на комбінаторних множинах невід'ємно пов'язані з комбінаторними многогранниками, що є опуклими оболонками таких множин, і їх властивостями. Властивості многогранника дають можливість звести розв'язання задач на дискретних комбінаторних множинах до їх розв'язання на неперервній допустимій множині. Тому виникає можливість застосовувати класичні методи неперервної оптимізації до розв'язання комбінаторних задач, що не завжди виправдано.

В загальному випадку ***задача комбінаторної оптимізації*** – це задача про максимізацію або мінімізацію функції при заданих обмеженнях та при умові, що на деякі, або навіть на всі, змінні накладена вимога цілочисельності. ЗКО виникає у величезній кількості практичних задач, одначе, багато важливих з теоретичної та прикладної точок зору класи ЗКО є NP-повними. Це означає, що для них невідомо та, швидше всього (при виконанні відомої гіпотези $P \neq NP$) не існує поліноміальних алгоритмів розв'язання. В силу вказаних причин гостро актуальною є проблема зменшення часу роботи алгоритмів розв'язання важкорозв'язних ЗКО, зокрема, проблема скорочення переліку можливих допустимих розв'язків в низці методів (гілок та границь, гілок та відсікань, динамічного програмування тощо). Одним з найпопулярніших та плідних підходів до скорочення переліку є поліедральний метод [284–286], якому присвячено безліч робіт. Цей підхід ґрунтується на використанні інформації про структуру многогранника (поліедра), який є опуклою оболонкою всіх допустимих розв’язків ЗКО. Важливість дослідження даного многогранника підкреслює, наприклад, той факт, що його фасетам (граням максимальної розмірності) відповідають найсильніші правильні відтинання. Структура многогранників допустимості ЗКО досліджувалась в різних роботах [87, 269]. Зокрема вивчались многогранники багатоіндексних транспортних задач [268, 272, 273] та багато інших. Геометричні аспекти теорії розрізів та метрик на поліедрах присвячена монографія M. M. Deza, M. Laurent [315]. В ній різноманітні результати, отримані для поліедрів у незалежних галузях математики, зв’язуються воєдино за допомогою теорії розрізів та метрик, зокрема, за допомогою понять розрізного конусу та розрізного многогранника. В. Н. Шевченко [269] сформулював гіпотезу, яка оголошує, що клас задач цілочисельного лінійного програмування, в яких мінори матриць обмежень обмежені константою d, є розв’язним за поліноміальний час. Добре відомо, що ця гіпотеза справедлива для d = 1. Частково гіпотеза підтверджена С. І. Веселовим та А. Ю. Чирковим [24, 327] для випадку d = 2. Як з теоретичної, так і з практичної точок зору видається дуже важливим побудова алгоритмів знаходження двоїстого опису поліедра (за його фасетами необхідно знайти вершини та конус рецесивних напрямків) і навпаки. Зазначимо, що на сьогодення питання про існування поліноміальних (від входу та виходу) алгоритмів, що розв’язують цю задачу, відкрите [297], та існують алгоритми, статус складності яких поки що невідомий, але які добре себе зарекомендували на практиці. Одним з них є відомий алгоритм Фур’є-Моцкіна [316]. Запропоновані різні модифікації та ефективні програмні реалізації даного алгоритму [107, 271, 305], а також його модифікації для побудови тріангуляцій [272]. Незважаючи на це, можливості для прискорення цих алгоритмів, напевне, не вичерпані. Близьким напрямком досліджень є друга фундаментальна задача, важлива і з теоретичної і з практичної точок зору, – вивчення структури граней поліедра (див., наприклад, 331) та його тріангуляції [272].

Видається важливим дослідження зв’язку між різними класами тріангуляцій точкових конфігурацій (регулярних, слабко регулярних, розгортуваних, симпліціально політопіальних тощо), зокрема, вивчення структури частково упорядкованої множини, побудованої на сімействі розглядуваних тріангуляційних класів відносно відношення включення. В останній час активно розвиваються алгоритми рішення опуклих поліедральних задач допустимості [289, 300]. Одним із цікавих обчислювальних підходів є зведення задачі допустимості для поліедра до задачі проекції на нормальний конус, породжений двоїстою системою нерівностей [174], що достатньо близьке до задачі проекції на двоїстий політоп.

На сьогоднішний день в галузі дослідження різних класів комбінаторних моделей, розробки нових методів їх розв'язання велика увага приділяється методам, що ґрунтуються на використанні структурних властивостей комбінаторних множин. Вивчення властивостей комбінаторних множин тісно пов'язане з теорією многогранників та графів. Використання інформації про структуру опуклої оболонки допустимих розв'язків – многогранників, яка є основою для багатьох методів, – один із самих успішних на сьогоднішній день підходів до розв'язання задач комбінаторної оптимізації. Але при розв'язанні таких задач виникають проблеми, пов'язані зі складністю математичних моделей, великим обсягом інформації та ін.

Прикладні задачі, що моделюються екстремальними дискретними задачами, досить часто мають велику вимірність, тому вони є досить складними з обчислювальної точки зору. Головна задача полягає в тому, щоб визначити таке значення аргументу, що належить деякій комбінаторній конфігурації, для якого цільова функція набуває глобального оптимуму. Тобто, необхідно розробити найбільш ефективний алгоритм, що ґрунтується на специфічних властивостях комбінаторних конфігурацій.

Для екстремальних задач комбінаторної оптимізації розроблялися і розробляються поліноміальні алгоритми знаходження оптимального розв'язку, що ґрунтуються на властивостях структури вхідних даних, але таких робіт, порівняно з методами, що ґрунтуються на частковому переборі варіантів, небагато. Одним із підходів до розв'язання такої проблеми є проведення дослідження і аналізу властивостей комбінаторних конфігурацій, на яких визначається цільова функція, що відображають комбінаторну природу задач. Аналізу і дослідженню комбінаторних конфігурацій як аргументу цільової функції, установленню зміни значень функції цілі в залежності від упорядкування елементів вибраної комбінаторної конфігурації і від специфіки структури вхідних даних у літературі достатньої уваги не приділяється. Але слід зазначити, що вивчення властивостей певних задач з метою виявлення їх характерних властивостей та їх використання для розв'язання поставленої задачі, дає можливість побудови нових підходів та розвитку відомих методів.

Отже, однією з важливих проблем в області дискретної оптимізації є виявлення властивостей комбінаторних конфігурацій в екстремальних задачах, використання яких дозволяло б установити закономірність зміни значень цільових функцій в залежності від упорядкування аргументу і від специфіки та структури множин комбінаторних конфігурацій.

На сьогоднішній день в області дослідження різних класів комбінаторних моделей і розробки нових методів їх розв'язання отримані істотні результати. Принципово вагомий внесок у розвиток дискретної, зокрема, комбінаторної оптимізації зробили такі закордонні вчені: М. Гері, Д. Джонсон, Х. Пападимитріу, К. Стайгліц, Р. Стенлі, Ф. Харарі, В. О. Ємелічев, В. М. Сачков, а також вчені з України: В. С. Михалевич, І. В. Сергієнко, Н. З. Шор, О. О. Ємець, В. О. Перепелиця, Ю. Г. Стоян, С. В. Яковлев та багато інших.

Розробка комплексного підходу до аналізу властивостей задач комбінаторної оптимізації охоплює широкий спектр досліджень комбінаторних функцій, комбінаторних многогранників, комбінаторних конфігурацій, як аргументу цільової функції, результати якого дають можливість покращити існуючі методи розв'язання таких задач і розробити нові методи знаходження оптимального розв'язку, що ґрунтуються на властивостях комбінаторних конфігурацій – актуальні проблеми комбінаторної оптимізації. Досить важливим в цьому аспекті є розгляд екстремальних задач на комбінаторних конфігураціях з використанням теорії графів.

Останнім часом графи і пов'язані з ними методи досліджень використовуються практично в усіх розділах сучасної математики, та особливо це стосується дискретної математики. Апарат теорії графів є потужним засобом моделювання задач управління дискретними процесами і системами. Граф, як математичне поняття, виявився простим, доступним та потужним засобом математичного моделювання складних систем.

На сьогодні створена досить розвинена теорія екстремальних задач на графах, яка включає дослідження структури властивостей різних класів задач, розробку алгоритмів їх розв'язання, отримання оцінок трудомісткості розв'язання та інших аспектів.

Дослідженням задач на графах присвячені роботи таких вчених як Ф. Харарі, О. Оре, І. В. Сергієнко, В. О. Ємелічев, О. О. Зиков, В. О. Перепелиця, Р. І. Тишкевич та ін.

Незважаючи на досить великі досягнення в області дискретної оптимізації, в процесі моделювання виникають класи екстремальних задач, для яких низка питань є ще не дослідженою.

Принципові труднощі, які виникають при моделюванні, пов'язані також із різними видами невизначеності. До їх числа відносяться: наявність багатьох критеріїв оцінки якості рішень, інтервальне задання параметрів задачі тощо. В цих умовах класичні методи виявляються не достатньо придатними для розв'язування задач.

Задачі оптимізації декількох функцій виникають при дослідженні багатьох теоретичних і прикладних проблем. В переважній більшості будь-яка прикладна задача оптимального проектування схем, технологічних пристроїв машин, конструкцій, складання сітьових графіків, планування і управління виробничою і комерційною діяльністю вимагає, щоб шуканий розв'язок знаходився з урахуванням багатьох критеріїв. Останні, як правило, суперечливі в тому розумінні, що якість порівнюваних альтернатив неможливо адекватно визначити одним комплексним критерієм, що демонструє деяку згортку початкових критеріїв. Отже, це означає, що апарат класичної однокритеріальної оптимізації є недостатнім для пошуку і досягнення ефективних розв'язків. В зв'язку з цим широко розвитку набуває дослідження і використання ширшої та більш загальної теорії – багатокритеріальної оптимізації.

Дослідження в області багатокритеріальної оптимізації в даний час особливо інтенсивно стимулюються практичними потребами і розвитком комп'ютерних інформаційних технологій, комп'ютерних мереж. Значний вклад у розвиток багатокритеріальної оптимізації в Україні внесли такі вчені як І. В. Сергієнко, В. С. Михалевич, Н. З. Шор, В. П. Шило, В. А. Трубін, Ю. Ю. Червак, В. А. Перепелиця та інші вчені.

Зокрема, в Запорізькому державному університеті під керівництвом В. А. Перепелиці його учнями проведені дослідження, які стосуються математичного моделювання задач в області економіки і техніки з використанням моделей дискретної багатокритеріальної оптимізації. Досить глибоко вивчені властивості комбінаторних оптимізаційних задач з векторним критерієм, питання їх складності, розв’язності, стійкості, алгоритмічні проблеми їх розв’язання.

Властивості розв’язності задач векторної оптимізації досліджені за допомогою алгоритму лінійної згортки критеріїв, який є найпоширенішим алгоритмом пошуку елементів множини Парето для векторних задач, а також завдяки широкому залученню методів теорії графів для розв’язування таких задач.

Вченими кафедри обчислювальної математики Ужгородського національного університету під керівництвом Ю. Ю. Червака досліджено низку нових моделей оптимального вибору за багатьма критеріями, які порівнюються між собою при оцінці альтернативних розв’язків задач оптимізації таким чином, що встановлюється апріорно часткова упорядкованість для кожних двох критеріїв, коли один з них важливіший, ніж другий, або ж обидва вони однаково важливі.

Цікаві результати одержані при застосуванні ідей лексикографічної оптимізації до розв’язання однокритеріальних задач дискретної оптимізації, зокрема, до задач, в яких частина або всі змінні є цілочисловими. Відомий загальний підхід до розв’язування задач у вигляді методу лексикографічних відтинань.

Але багато задач проектування, планування, розміщення, управління та інші описуються за допомогою різних моделей багатокритеріальної оптимізації, розв’язки яких мають комбінаторні властивості, зокрема, переставні, сполучні та інші, тобто пошук розв’язків здійснюється на деякій комбінаторній конфігурації.

Як відомо, більшість комбінаторних оптимізаційних задач можуть бути зведені до задач цілочисельного програмування, але це не завжди виправдано, оскільки при цьому втрачається можливість врахування комбінаторних властивостей розв’язків задач.

Дана робота є продовженням досліджень в області екстремальних задач дискретної, зокрема комбінаторної оптимізації, а також задач оптимізації за умови багатокритеріальності, що виникають при дослідженні багатьох теоретичних і прикладних проблем.

У монографії розглядається нова і актуальна задача, яка об’єднує в собі проблему багатокритеріальності і комбінаторні властивості розв’язків; розроблено новий метод розв’язування екстремальних задач на комбінаторних конфігураціях, який називається методом направленого структурування. Основна ідея запропонованого методу перекликається з відомим методом послідовного аналізу варіантів, суть якого описано в оглядовій частині. Але новий метод призначений безпосередньо для комбінаторних задач. Як відомо, у більшості задач на комбінаторних конфігураціях постає необхідність перераховувати велику кількість варіантів, порівнянну з факторіальною величиною. Тому, метод об'єднує засоби комбінаторного аналізу та теорії графів.

В роботі наведено приклади застосування нового методу з застосуванням теорії графів для розв'язування екстремальних комбінаторних задач з лінійною та дробово-лінійною функціями.

РОЗДІЛ 1. КОМБІНАТОРНІ КОНФІГУРАЦІЇ ТА ЇХ ВЛАСТИВОСТІ

В даному розділі розглядаються різні комбінаторні конфігурації, визначення яких може бути сформульованим на основі загального поняття конфігурації, представленого в термінах відображення множини.

1.1. Визначення комбінаторних конфігурацій, різні види, та їх обмеження

Під множиною розуміється сукупність предметів і понять, об'єднаних деякою загальною властивістю. Іноді замість слова «множина» вживають терміни: сукупність, сімейство й т. д. Множина складається з елементів, і запис $x \in X$ означає, що елемент x належить множині X; в іншому випадку пишуть $x \notin X$, і це означає, що елемент x не належить множині X. Якщо кожний елемент $x \in X$ має властивість $x \in Y$, то говорять, що X є підмножина Y і записують $X \subseteq Y$. Множини X та Y рівні, якщо $X \subseteq Y$ і $Y \subseteq X$, що відповідає запису $X = Y$. X є власною підмножиною Y, $X \subset Y$, якщо $X \subseteq Y$ і $X \neq Y$. Будь-яка множина містить як підмножину порожню множину, що позначається символом $\varnothing$.

Представлення множини можна задати або виписуванням її елементів, або шляхом вказівки їх загальних властивостей. Використовуючи другий спосіб опису множин, визначимо для них так звані булеві операції.

Розглянемо деякі з них:

1. Об'єднання

$$X \bigcup Y = \{x : x \in X \quad \text{або} \quad x \in Y\}$$

тобто $X \bigcup Y$ є множина елементів, які належать принаймні одній з множин X і Y.

2. Перетин

$$X \bigcap Y = \{x : x \in X\,,\ x \in Y\}.$$

3. Різниця

$$X \setminus Y = \{x : x \in X,\ x \notin Y\}.$$

4. Доповнення множини *X* відносно *Y*, $X \subseteq Y$

$$\bar{X} = Y \setminus X.$$

Сукупність всіх підмножин множини *X*, включаючи порожню множину, називається булеаном *X* і позначається 2^x. Елементи $X_1, X_2, \ldots, X_r$ булеана 2^x утворюють розбиття множини *X*, якщо $X_i \neq \varnothing$, $i = 1, 2, \ldots, r$ і

$$X = X_1 \bigcup X_2 \bigcup \ldots \bigcup X_r;\ X_i \bigcap X_g = \varnothing,\ i \neq g. \qquad (1.1)$$

Множини $X_1, X_2, \ldots, X_r$ називають блоками розбиття. Сукупність всіх упорядкованих пар (x,y) таких, що $x \notin X$, називається декартовим добутком множин *X* і *Y* і позначається $X \times Y$ тобто

$$X \times Y = \{(x,y) : x \in X, y \in Y\}.$$

Аналогічно визначається декартовий добуток *r* множин

$$X_1 \times X_2 \times \ldots X_r = \{(x_1, x_2, \ldots, x_r) x_1 \in X_1, x_2 \in X_2, \ldots, x_r \in X_r\},$$

а при збігу множин – декартова степінь $X^{(r)} = X \times X \times \ldots \times X$.

Нехай *X* – скінченна множина, а $|X|$ – число її елементів. Сформулюємо очевидне аксіоматичне правило, що лежить в основі багатьох комбінаторних обчислень.

Правило суми. Якщо *X* – скінченна множина і

$$X = X_1 \bigcup X_2 \bigcup \ldots \bigcup X_r,\ X_i \in 2^x, i = 1, 2, \ldots, r, \text{ то}$$
$$|X| \leq |X_1| + |X_2| + \ldots + |X_r|,$$

причому рівність досягається тоді, коли $X_1, X_2, \ldots, X_r$ утворять розбиття *X*.

Приведемо друге просте правило, що використовується в комбінаторному аналізі.

Правило добутку. Для скінченних множин $X_1, X_2, \ldots, X_r$

$$\left|X_1 \times X_2 \times \ldots \times X_r\right| = \left|X_1\right| \cdot \left|X_2\right| \cdot \ldots \cdot \left|X_r\right|.$$

2. Бінарні відповідності й бінарні відношення *t*. Усяка множина пар $R \subseteq X \times Y$ називається бінарною відповідністю на множинах X і Y. Якщо $(x,y) \in R$ то x і y назвемо проекціями (x,y) на X і Y відповідно й будемо записувати в таким способом:

$$x = \pi_1(x,y), \quad y = \pi_2(x,y).$$

Для бінарної відповідності $R \subseteq X \times Y$ можна визначити її проекцію на X і Y:

$$\pi_1(R) = \{x : x = \pi_1(x,y),\ (x,y) \in R\}$$

$$\pi_2(R) = \{y : y = \pi_2(x,y),\ (x,y) \in R\}.$$

Проекції $\pi_1(R), \pi_2(R)$ іноді називаються відповідно областю визначення й областю значення R. Образом елемента $x \in X$ при відповідності R будемо називати множину

$$\delta_1(x;R) = \{y : y \in Y,\ (x,y) \in R\}.$$

Аналогічно прообразом елемента $y \in Y$ при відповідності R називається множина

$$\delta_1(y;R) = \{x : x \in R,\ (x,y \in R)\}.$$

Бінарна відповідність $\varphi \subseteq X \times Y$ називається функціональною, якщо образ кожного елемента $x \in X$ містить рівно один елемент.

Для множин $X = (x_1, x_2, \ldots, x_n), Y = \{y_1, y_2, \ldots, y_m\}$ бінарній відповідності $R \subseteq X \times Y$ в множині можна поставити матрицю $A = \left\|a_{ig}\right\|, i = 1, 2, \ldots, n, g = 1, 2, \ldots, m.$

І навпаки, будь-якому розбиттю X відповідає відношення еквівалентності, класи якого збігаються з блоками зазначеного розбиття. Множина всіх класів еквівалентності називається фактор-множиною по даному відношенню еквівалентності.

Бінарне відношення R на множині X називається відношенням часткового порядку, якщо воно рефлексивне, антисиметричне й транзитивне. Множина X називається в цьому випадку частково впорядкованою множиною. Якщо R – відношення часткового порядку і xRx', то звичайно пишуть: $x \le x'$.

Відношення часткового порядку R, що задовольняє умові дихотомії, будемо називати відношенням лінійного порядку або просто відношенням порядку. Запис $x \le x'$ використовуємо для позначення того, що x і x' зв'язано відношенням R лінійного порядку: xRx'.

На множині X можна визначити відношення строгого часткового (лінійного) порядку – $\prec(\prec)$, вважаючи що $x \prec x'(x \prec x')$, якщо $x \le x'(x \le x')$ і $x \ne x'$.

Ясно, що на скінченій множині X завжди можна задати відношення строгого порядку. Кожне таке відношення визначає перестановку елементів цієї множини, і число таких відношень дорівнює $|X|!$.

На булеані 2^x скінченої множини X можна задати відношення часткового порядку, вважаючи $X \le X'$ тоді й тільки тоді, коли $X \subseteq X'$ для всіх X, $X' \in 2^x$. У свою чергу на декартовій степені X^r скінченої множини X із заданим строгим лінійним порядком можна встановити відношення лінійного порядку, при якому

$$\left(x_1, x_2, \ldots, x_r\right) \prec \left(x_1, x_2, \ldots, x_r\right), \left(x_1, x_2, \ldots, x_r'\right) \in X^{(r)},$$

якщо для найменшого індексу i із властивістю $x_i \ne x_i'$ має місце $x_i \prec x_i'$. Такий порядок є лексикографічним і використовується для впорядкування слів у словниках.

Відображення. Закон φ, відповідно до якого кожному елементу $x \in X$ ставиться у відповідність єдиний елемент $\varphi(x) \in Y$, називається однозначним відображенням множини X в множину Y або функцією.

При $X = Y$ бінарну відповідність $R \subseteq X \times X$ будемо називати бінарним відношенням на множині *X*. Прикладом бінарного відношення на множині X є відношення рівності $\varDelta_x \{(x,x): x \in X\}$.

Бінарному відношенню *R* на скінченній множині *X* можна поставити у відповідність геометричний об'єкт, що називається орієнтованим графом або діаграмою. Кожному елементу $x \in X$ ставиться у відповідність точка на площині, що називається вершиною. Якщо $(x,x') \in R,$ то точки, визначені x і x', з'єднуються стрілкою, що виходить з x і заходить в x', яку називають дугою. Сукупність вершин і дуг, побудованих таким чином, являє собою діаграму $\varGamma(X,R)$ відношення *R*.

Для бінарного відношення R на множині X домовимося, що якщо $(x,x') \in R,$ то цей факт записується таким способом: xRx'. Використовуючи це позначення, сформулюємо ряд властивостей, якими можуть володіти бінарні відношення:

1) рефлексивность: xRx для всіх $x \in X;$
2) антирефлексивность: $R \bigcap \varDelta_x = \varnothing;$
3) симетричність: із xRx' випливає $x'Rx;$
4) антисиметричність: з xRx' і $x'Rx$ випливає $x = x';$
5) транзитивність: з xRx', $x'Rx''$ випливає $xRx'';$
6) дихотомія: або xRx', або $x'Rx, x,x' \in X$.

Бінарне відношення *R* на множині *X* називається відношенням еквівалентності, якщо воно одночасно рефлексивне, симетричне і транзитивне. Якщо *R* – відношення еквівалентності і $xRx',$ то звичайно пишуть $x \approx x'$. Множина елементів $K(x) = \{x' : x' \approx x\}$ назвемо класом еквівалентності, що містить $x,x' \in X$. Класи еквівалентності утворюють розбиття множини *X*.

Закон φ, згідно якого кожному елементу $x \in X$ ставиться у відповідність єдиний елемент $\varphi(x) \in Y,$ називається однозначним відображенням множини X в множину *Y* або функцією, що визначена на *X* і приймає значення в *Y*. Цьому відображенню відповідає запис $\varphi: X \to Y$. Очевидно, що однозначне відображення $\varphi: X \to Y$ являє собою функціональну бінарну відповідність $\varphi \subseteq X \times Y$. Многозначне відображення ψ множини X в множину Y визначимо як закон, за яким кожному елементу $x \in X$ ставиться у відповідність множина $\psi(x) \subseteq Y$, причому не виключається, що $\psi(x) = \varnothing$. Ясно, що многозначному відображенню ψ множини X у Y відповідає деяка бінарна відповідність $\psi \subseteq X \times Y$. Надалі будемо використовувати в основному однозначні відображення.

У зв'язку із цим зупинимося на властивостях однозначних відображень, для зручності опускаючи постійне згадування про однозначність.

Найбільш вживані дві форми запису відображень: функціональна й дворядкова.

Функціональний запис: $y = \varphi(x)$ означає, що при відображенні $\varphi: X \to Y$ елемент $x \in X$ переходить в елемент $y \in Y$. Для скінчених множин X і Y часто використовується дворядковий запис. Якщо $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_m\}$, то дворядковий запис, що відповідає $\varphi: X \to Y$ має вигляд

$$\varphi = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ \varphi(x_1) & \varphi(x_2) & \ldots & \varphi(x_n) \end{pmatrix}.$$

Розглянемо 0,1-матрицю $A = \|a_{ig}\|$, $i = 1, 2, \ldots, n, g = 1, 2, \ldots, m,$

де $a_{ig} \begin{Bmatrix} 1, & y_{g=\varphi(x_i)} \\ 0, & y_g \neq \varphi(x_1) \end{Bmatrix}$.

Матрицю A назвемо матрицею інцидентності відображення φ множини $X = \{x_1, x_2, \ldots, x_n\}$ в множину $Y = \{y_1, y_2, \ldots, y_m\}$. У силу однозначності відображення φ матриця A в кожному рядку і в кожному стовпцю має тільки одну одиницю. Такі 0, 1-матриці називають елементарними.

Два відображення $\varphi_1 : X_1 \to Y_1, \varphi_2 : X_2 \to Y_2$ будемо вважати рівними, якщо $X_1 = X_2, Y_1 = Y_2$ або $\varphi_1(x) = \varphi_2(x)$ для всіх $x \in X_1$. Відображення $\varphi' : X' \to Y$ є звуженням, або обмеженням на X' відображення $\varphi: X \to Y$ якщо $X' \subseteq X$ і $\varphi'(x) = \varphi(x)$ для кожного $x \in X'$.

Множина $\varphi^{-1}(y) = \{x : \varphi(x) = y, x \in X\}$ називається повним прообразом елемента y при відображенні $\varphi: X \to Y$. Повний прообраз елемента y збігається із прообразом цього елемента при функціональній бінарній відповідності $R \subseteq X \times Y$, що відповідає відображенню φ.

Відображення $\varphi: X \to Y$ сюр'єктивне, якщо для кожного $y \in Y$ існує таке $x \in X$, що $y = \varphi(x)$. У цьому випадку також говорять, що φ є відображення X на Y. Очевидно, що при сюр'єктивному відображенні повний прообраз кожного елемента $y \in X$ не є порожньою множиною.

Відображення $\varphi: X \to Y$ ін'єктивне, якщо для будь-яких $x, x' \in X$ таких, що $x \neq x'$, треба $\varphi(x) \neq \varphi(x')$.

Повний прообраз кожного елемента $y \in Y$ при ін'єктивному відображенні містить не більше одного елемента.

Відображення $\varphi: X \to Y$ бієктивне, якщо воно одночасно сюр'єктивне і ін'єктивне. Такі відображення називають також взаємно однозначними.

Якщо $X = Y$, то бієктивне відображення $\varphi: X \to X$ називається підстановкою. Кожній підстановці кінцевої множини X відповідає квадратна матриця інцидентності порядку $|X|$, що має в кожному рядку й у кожному стовпці по одній і тільки одній одиниці. Такі матриці називаються підстановочні.

Закони композиції. Законом композиції або операцією (бінарною) на множині X будемо називати відображення $f: X \times X \to X$. Якщо $f(x,y) = z$, де $x, y, z \in X$, то будемо записувати цей факт таким чином: $xTy = z$.

Закон композиції T називається асоціативним, якщо для будь-яких $x, y, z \in X$ $(xTy)Tz = xT(yTz)$.

Якщо закон композиції задовольняє умові $xTy = yTx$ для будь-яких $x, y \in X$, то він називається комутативним.

Закон композиції *T* є дистрибутивним щодо закону $\perp$, якщо для будь-яких $x, y, z \in X$ виконуються співвідношення

$$\begin{aligned} xT\left(y \perp z\right) = \left(xTy\right) \perp \left(xTz\right) \\ \left(y \perp z\right)Tx = \left(yTx\right) \perp \left(zTx\right) \end{aligned}.$$

Стосовно закону композиції *T* введемо поняття одиничного елемента і елемента, оберненого до даного.

Елемент $e \in X$ називається одиничним або нейтральним щодо закону композиції *T,* якщо для будь-якого $x \in X : xe = ex = x$.

Якщо такий елемент існує, то він єдиний. Нехай закон композиції *T* має одиничний елемент. Тоді елемент x^{-1} називається оберненим або симетричним до елемента $x \in X$ щодо цього закону, якщо

$$xx^{-1} = x^{-1}x = e .$$

Якщо – *X* скінчена група, то величину $\left|X\right|$ будемо називати її порядком. Нехай *X* і *Y* – скінченні групи довільних порядків із законами композиції *T* і $\perp$ відповідно і $\varphi : X \to Y$ – відображення, при якому для будь-яких $x, x' \in X$ має місце рівність

$$\varphi\left(xx'\right) = \varphi\left(x\right) \perp \left(x'\right).$$

Таке відображення φ називається гомоморфізмом групи *X* в групу *Y*. Якщо $-\varphi$ бієктивне відображення, то гомоморфізм називається ізоморфізмом.

Непуста підмножина *Y* групи *X* із законом композиції *T* називається підгрупою, якщо виконані наступні умови:

1) з того, що $y \in Y,$ слідує $y^{-1} \in Y;$

2) з умови $y \in Y, y' \in Y$ слідує $y', y \in Y$.

Якщо в підмножині $Y \subset X$ композиція будь-яких двох елементів з *Y* також належить *Y*, то говорять, що *Y* замкнута множина щодо даного закону композиції.

Якщо *X* – скінчена група, а $x_1, x_2, \ldots, x_{k-1}$ – представники всіх правих суміжних класів по підгрупі *Y*, не вважаючи самого *Y*, то можна представити розкладання у вигляді

$$X = Y\bigcup Yx_1\bigcup\ldots\bigcup Yx_{k-1}.$$

Якщо в булеані 2^x як закон $\perp$ взяти операцію об'єднання $\bigcup$, у якості *T* – операцію перетину $\bigcap$ підмножин множини *X*, до зазначених операцій додати операцію доповнення, то одержимо булеву алгебру.

Множина цілих чисел *X*, у якому законами композиції є звичайне додавання й множення, утворить кільце.

Говорять, що числа $a,b\in Z$ порівнювані за модулем m, якщо їх різниця $a-b$ ділиться на *m*. Цей факт записують так: $a\equiv b(mob\ m)$. Порівняння за модулем є відношення еквівалентності, а відповідні класи множин називаються класами лишків за модулем *m*. Кожний із класів містить у точності одне із чисел $0,1,\ldots,m-1$, які є представниками класів і утворять повну систему найменших недоданих лишків за модулем *m*.

Позначимо через K_i – клас, представником якого є число $i, 0\le i\le m-1$.

На множині класів лишків $K_0, K_1,\ldots, K_{m-1}$ можна ввести операції додавання та множення $+$ і $\cdot$. Ці операції над зазначеними класами будемо визначати таким способом:

$K_i + K_g = K_l$, якщо $i+g=l(mod\, m)$,

$K_i\cdot K_g = K_l$ якщо $i\cdot g\equiv l(mod\, m)$.

У результаті одержуємо кільце класів лишків за модулем m, що будемо позначати через Z_m.

У кільці нейтральні елементи щодо законів композиції $\perp$ й *T* називаються відповідно нулем і одиницею.

Якщо сукупність відмінних від нуля елементів щодо закону композиції *T* утворить абелеву групу, то в цьому випадку кільце називається полем.

Поле, що містить скінчене число елементів, називається скінченим. Якщо *p* – просте число, то сукупність класів лишків за модулем *p* утворить скінчене поле, що містить *p* елементів.

На основі вище введених понять розглянемо властивості різних видів комбінаторних конфігурацій.

1.2. Властивості різних видів комбінаторних конфігурацій

При розв'язанні комбінаторних задач часто використовуються такі добре відомі конструкції з елементів скінченої множини, як сполучення, розміщення, перестановки тощо. З точки зору комбінаторики, сполученням об'єму m називається підмножина з m елементів вихідної множини $Y=\{y_1, y_2, \ldots, y_n\}$. Якщо для вибраних m елементів є істотним порядок, то говорять про розміщення m елементів. При $m=n$ розміщення являє собою перестановку елементів множини Y. В результаті одержимо сполучення, розміщення, перестановки з повтореннями.

Уже для цих найпростіших комбінаторних конструкцій виникає необхідність формалізації їх визначення з метою уникнути словесних накопичень і плутанини. З ускладненням конструкцій така необхідність стає актуальнішою. Згадана формалізація може бути здійснена у більшості класів випадків шляхом введення поняття конфігурації.

Більш строго формалізація комбінаторних об'єктів може бути здійснена на основі запропонованого в [206, 207] поняття комбінаторної конфігурації.

В означенні комбінаторної конфігурації важливу роль грає формалізація поняття нерозрізненості об'єктів. В даний час добре відомий підхід до визначення нерозрізненості в комбінаторному аналізі, що використовує поняття класів еквівалентності щодо деякої групи підстановок G, яка діє на вхідній множині X. При такому підході множина нерозрізнених об'єктів, або, як часто говорять, задача перерахування, зводиться до визначення коефіцієнтів деякого поліному, що залежить від циклових класів групи G. Даний метод, що одержав у комбінаторному аналізі назву теорії перерахування Пойа, бере свій початок з робіт І. Редфілда й Г. Пойа, а його подальший розвиток можна знайти в роботі де Брейна. Слід зазначити, що більшу частину теорії перерахування Пойа можна застосувати в розробці підходів до розв'язання задач перерахування і насамперед у теорії графів. Однак у цілому ряді важливих випадків застосування методу Пойа зустрічає технічні складності й стає малоефективним. Це насамперед відноситься до таких відомих комбінаторних схем, як розміщення, сполучення, перестановки з різноманітними властивостями, заповнення комірок предметами з обмеженнями на ємність комірок і вказівкою критеріїв розпізнавання заповнень і т. п.

Тому, одним із важливих понять в даному розділі є визначення загальної комбінаторної схеми та її використання до розв'язування екстремальних задач. Ця схема як часткові випадки охоплює відомі комбінаторні схеми, у тому числі згадувані вище розміщення, перестановки, сполучення, розбиття і т. п. Вона дозволяє сформулювати методику побудови часткових комбінаторних схем, об'єднаних деякими загальними ознаками.

Для формалізації комбінаторних схем використовується поняття відображення однієї скінченої множини X у іншу Y, шляхом введення поняття конфігурації. За цих умов кожному заповненню m предметів в n комірках ставиться у взаємно однозначну відповідність деяка конфігурація $\varphi: X \to Y$, $|X| = m,\ |Y| = n,$ а розрізнення заповнення точно визначається властивостями конфігурації φ. Можливий і інший підхід до формалізації комбінаторних схем розглянутого вигляду, що також використовує поняття відображення. Розглянемо суть цього підходу на прикладі урнової схеми.

Розглянемо модель заповнення m різних предметів в n різних комірках. У цьому випадку кожне заповнення знову визначається відображенням $\varphi: X \to Y,\ |X| = m,\ |Y| = n,$ причому на φ не накладено ніяких обмежень. Для побудови моделі, у якій комірки різні, а предмети нерозрізнені(однакові), на множині заповнень, що відповідають вихідній моделі, можна задавати відношення еквівалентності, при якому два заповнення еквівалентні, якщо вони відрізняються тільки нумерацією предметів. Ця еквівалентність індукує відповідне відношення еквівалентності на множині відображень, що відповідають заповненням.

Наступний за формалізацією етап у розв'язанні комбінаторної задачі полягає в побудові твірної функції для перерахування різних об'єктів відповідно до деякої характеристики, що називається іноді вагою.

Типовими характеристиками об'єктів є так звані первинні й вторинні специфікації відповідних їм відображень. Можна сказати, що специфікації визначають, у деякому розумінні, склад елементів, що є образами відображення з урахуванням повторень цих елементів. Нехай кожна із груп *G* і *H* може збігатися або з одиничною, або із симетричною групою відповідного ступеня. Використовуючи як характеристики первинні, або вторинні специфікації, у кожному з чотирьох випадків, що при цьому виникають, можна вказати спосіб побудови виробничих функцій для перерахування нееквівалентних відображень при різноманітних обмеженнях на ці специфікації. Дану формалізацію певного класу комбінаторних задач разом з методикою побудови відповідних виробничих функцій називають загальною комбінаторною схемою.

Дамо визначення комбінаторної конфігурації, що є основним поняттям в комбінаторній схемі.

Нехай $X=\{1,2,\ldots,m\}$, $Y=\{y_1,y_2,\ldots,y_n\}$ − скінчена лінійно впорядкована множина, і нехай на *Y* задано строгий лінійний порядок: $y_1 < y_2 < \ldots < y_n$.

Будемо говорити, що задано відображення φ множини *X* в множину *Y*, якщо кожному елементу $x \in X$ ставиться у відповідність єдиний елемент $y \in Y$. При відображенні φ відповідність між x і y записується рівністю $y=\varphi(x)$, а самому відображенню відповідає запис $\varphi: X \to Y$. Множина *X* є область визначення відображення, а *Y* – область його значень.

Визначення 1.1. Відображення $\varphi: X \to Y$, що задовольняє деякому комплексу обмежень Λ, будемо називати конфігурацією.

Отже, під комбінаторною конфігурацією розуміється відображення $\varphi: X \to Y$, яке задовольняється певній системі обмежень Λ.

На основі цього підходу побудовані комбінаторні конфігурації, які відповідають найпростішим комбінаторним об’єктам: розміщенням, перестановкам, комбінаціям, розбиттям та іншим об’єктам, утвореним на основі урнової схеми [206].

Комплекс обмежень Λ, якому задовольняє відображення φ, визначає деякий клас конфігурацій, що відповідають умовам на комбінаторній конструкції в розглянутій задачі.

Розглянемо деякі властивості відображень. Відображення $\varphi: X \to Y$ називається сюр’єктивним, якщо будь-який елемент $y \in Y$ має хоча б один прообраз при цьому відображенні.

Іншими словами, для кожного $y \in Y$ існує $x \in X$ такий, що $y = \varphi(x)$. У цьому випадку іноді говорять, що φ відображає *X* на *Y*. Якщо φ – сюр’єктивне відображення, то $\varphi(X) = Y$. Для скінчених множин *X* і *Y* сюр’єктивність відображення $\varphi : X \to Y$ означає, що $|X| \geq |Y|$.

Відображення $\varphi : X \to Y$ називається ін’єктивним, якщо для будь-якого $y \in Y$ його повний прообраз $\varphi^{-1}(y)$ містить не більше одного елемента. Іншими словами, відображення $\varphi : X \to Y$ ін’єктивне, якщо для будь-яких $x \neq x'$, $x, x' \in X$, $\varphi(x) \neq \varphi(x')$. Якщо *X* й *Y* скінченні, то ін’єктивність відображення $\varphi : X \to Y$ означає, що $|X| \leq |Y|$.

Відображення $\varphi : X \to Y$ називається бієктивним, якщо воно одночасно сюр’єктивне й ін’єктивним. Із сюр’єктивності слідує, що $\left|\varphi^{-1}(y)\right| \geq 1$ для кожного $y \in Y;$ з ін’єктивності випливає умова $\left|\varphi^{-1}(y)\right| \leq 1$ для кожного $y \in Y$. Отже, бієктивність відображення φ означає, що $\left|\varphi^{-1}(y)\right| = 1$ для будь-якого $y \in Y,$ тобто умова $y = \varphi(x)$ для кожного $y \in Y$ однозначно визначає єдине значення $x \in X$. У цьому випадку говорять, що бієктивне відображення φ встановлює взаємнооднозначну відповідність між множинами *X* й *Y*. Коли *X* й *Y* скінченні, це означає рівність $|X| = |Y|$.

На основі зазначених властивостей відображень опишемо побудову класів конфігурацій, що відповідають найпростішим комбінаторним конструкціям: розміщенням, сполученням, перестановкам, заповненням комірок елементами; тобто дамо строгі визначення цих конструкцій, якими будемо користуватися надалі. Поняття конфігурацій, пов’язаних з вибором і розташуванням деяких об’єктів відповідно до певних правил.

Нехай $\Lambda = \varnothing,$ тобто відсутні будь-які обмеження на відображення $\varphi : X \to Y$. Тоді кожна конфігурація φ визначає комбінаторну конструкцію, названу розміщенням з необмеженими повтореннями об’єму m з n різних елементів або m-перестановкою. Очевидно, що число таких конфігурацій дорівнює

$$\bigcup(n,m) = n^m.$$

Нехай $\Lambda = I$, де I − множина ін’єктивних відображень, тобто множина таких відображень, для яких заборонено два різні елементи $x_i, x_j \in X$ відображати в один елемент $y_i \in Y$. У цьому випадку конфігурація $\varphi : X \to Y$ називається розміщенням об’єму m з n різних елементів або m-перестановкою.

Число таких конфігурацій дорівнює $A(n,m) =$ $= n(n-1)\ldots(n-m+1) = (n)_m$, $m \le n$, де $(n)_m = n!/(n-m)!$ і $A(n,m) = 0, m > n$. Числа $A(n,m)$ задовольняють рекурентному співвідношенню

$$A(n,m) = A(n-1,m) + mA(n-1,m-1).$$

При $m = n$ кожній конфігурації відповідає перестановка, і їх число дорівнює $P(n) = n!$.

Множину перших k натуральних чисел позначимо N_k, а $N_0^k = N_k \bigcup \{0\}$.

Якщо n елементів містять q_i i-го виду, де $i \in N_k$, $q_1 + q_2 + \ldots + q_k = n$ і елементи одного виду ідентичні, то кожній перестановці цих елементів відповідає конфігурація $\varphi : X' \to Y'$, де $X' = \{1, 2, \ldots, n\}$, $Y' = \{y_1, y_2, \ldots, y_k\}$, і y_i використовується як образ q_i раз. Число таких перестановок дорівнює

$$P(q_1, q_2, \ldots, q_k) = \frac{n!}{q_1!q_2!, \ldots, q_k!}.$$

Визначимо конфігурацію як Λ однозначне відображення, що задовольняє комплексу умов $\varphi : X \to Y$, де $X = \{1, 2, \ldots, m\}$, Y − скінчена множина. Далі розглянемо конфігурації σ, для яких усякі обмеження відсутні, тобто $\Lambda = \varnothing$.

Позначивши M множину строго монотонних функцій, тобто таких, що $\varphi(i) < \varphi(j)$, коли $i < j$. Тоді маємо при $\Lambda = M$ відображення сполучень, оскільки умова $\Lambda = M$ визначає заборону монотонності елементів, а отже відсутність відношення порядку.

Нехай σ – конфігурація, що представляє собою відображення множини $X = \{1, 2, \ldots, m\}$ в множину $A = \{a_1, a_2, \ldots, a_n\}$. Конфігурація σ може бути записана у вигляді

$$\sigma = \begin{pmatrix} 1 & 2 & \ldots & m \\ \sigma(1) & \sigma(2) & \ldots & \sigma(m) \end{pmatrix} = \begin{pmatrix} 1 & 2 & \ldots & m \\ a_{i_1} & a_{i_2} & \ldots & a_{i_m} \end{pmatrix},$$

де $\sigma(j) = a_{i_j}$, $j \in N_m$. Іноді зручно конфігурацію σ записувати у вигляді m-мірного вектора з компонентами з множини $A: \sigma = (\sigma(1), \sigma(2), \ldots, \sigma(m))$ або $\sigma = \left(a_{i_1}, a_{i_2}, \ldots, a_{i_m}\right)$.

Дамо визначення двох зв'язаних між собою характеристик конфігурації σ, названих первинною й вторинною специфікаціями.

Визначення 1.2. Вираз $[\sigma] = \left[a_1^{\alpha_1}, a_2^{\alpha_2}, \ldots, a_n^{\alpha_n}\right]$, $\alpha_1 + \alpha_2 + \ldots + \alpha_n = m$ називається первинною специфікацією конфігурації σ, якщо вектор $\left(a_{i_1}, a_{i_2}, \ldots, a_{i_m}\right)$ має α_j елементів a_i із множини A, де $j \in N_n$. Число α_j називається a_i – показником первинної специфікації σ.

Якщо множину A піддамо перетворенню, що полягає в розмноженні елемента a_i в α_j в ідентичних екземплярах, де $\alpha_1 + \alpha_2 + \ldots + \alpha_n = m$, то в результаті одержимо мультимножину A' образів відображення $\sigma: X \to A$. Зазначимо, що мультимножина є об'єднання не обов'язково різних елементів; її можна вважати множиною, в якій кожному елементу поставлено у відповідність додатне ціле число, що називається кратністю. Скінчену множину A' можна записувати в наступному вигляді:

$$A = \{a_1, a_2, \ldots, a_n\},$$

де $a_1, a_2, \dots, a_n$ – обов’язково різні елементи A'. Потужність множини A' позначимо через $|A'|$, і для виписаної вище множини $|A'| = n$. Якщо A' – скінчена множина, то будемо записувати її в наступному вигляді:

$$A = \left\{ \underbrace{a_1, a_1, \dots, a_1}_{\alpha_1\ раз}, \underbrace{a_2, a_2, \dots, a_2}_{\alpha_2\ \text{раз}}, \dots, \underbrace{a_n, a_n, \dots, a_n}_{\alpha_n\ раз} \right\} =$$

$$= \{\alpha_1 \cdot a_1, \alpha_2 \cdot a_2, \dots, \alpha_n \cdot a_n\}.$$

Тут всі a_i – різні і α_i – кратність елементу a_i. В цьому випадку потужність A' рівна

$$|A'| = \sum_{i=1}^{n} \alpha_i \ .$$

Склад елементів мультимножини A' однозначно визначається вектором $(\alpha_1 + \alpha_2 + \dots + \alpha_n)$. Будемо говорити, що мультимножина A' має первинну специфікацію $\left[a_1^{\alpha_1}, a_2^{\alpha_2}, \dots, a_n^{\alpha_n}\right]$, тому що вектор $\left(a_{i_1}, a_{i_1}, \dots, a_{i_m}\right)$, що відповідає конфігурації σ, та первинній специфікації σ і A' збігаються.

Визначимо відмінність між первинними специфікаціями відображення σ й мультимножиною його образів. Нехай серед чисел $\alpha_1, \alpha_2, \dots, \alpha_n$, що визначають первинну специфікацію конфігурації σ є β_0 нулів, β_1 одиниць, β_2 двійок і т. д. Тоді, вираз

$$\left[[\sigma]\right] = \left[\left[0^{\beta_0} 1^{\beta_1} 2^{\beta_2} \dots m^{\beta_m}\right]\right],$$

$\beta_0 + \beta_1 + \dots + \beta_m = n, \qquad \beta_1 + 2\beta_2 + \dots + m\beta_m = m,$ називається вторинною специфікацією конфігурації σ. Число β_j називається j-показником вторинної специфікації. Іноді 0-показник вторинної специфікації σ можна опускати, записуючи $\left[\left[\sigma\right]\right]=\left[\left[1^{\beta_1}2^{\beta_2}\ldots m^{\beta_m}\right]\right]$, $\beta_0+\beta_1+\ldots+\beta_m=n$, $\beta_1+2\beta_2+m\beta_2=m$.

Аналогічним чином можна говорити про те, що мульти-множина образів відображення має вторинну специфікацію $\left[\left[0^{\beta_0}1^{\beta_1}2^{\beta_2}\ldots m^{\beta_m}\right]\right]$, $\beta_0+\beta_1+\ldots+\beta_m=n$, $\beta_1+2\beta_2+\ldots+m\beta_m=m$, якщо число елементів множини A', що зустрічається в A' i раз, дорівнює $\beta_i\left(i=0,1,\ldots,m\right)$. Множину всіх відображень $\sigma: X\to A$ будемо позначати A^x. Ясно, що між A^x і m-й декартовим ступенем $A^{(m)}$ існує бієктивна відповідність.

У дискретній математиці важливу роль грають комбінаторні методи, необхідність у застосуванні яких виникає щораз, коли цікавляться існуванням відповідних алгоритмів та числом способів розташування елементів деякої множини відповідно до заданих правил. Кожне таке розташування визначає комбінаторну конфігурацію, яку можна розглядати як відображення однієї множини в іншу з деякими обмеженнями, обумовленими конкретними вимогами до задачі.

Якщо обмеження носять складний характер, то для відповідних комбінаторних конфігурацій необхідно з'ясувати умови існування й знаходити методи їх конструювання.

Для подальшого викладу матеріалу розглянемо поняття елементу довільної комбінаторної конфігурації як вершини деякого графа.

Як відомо, під комбінаторним об'єктом розуміють підмножину заданої дискретної множини, що задовольняє деяким властивостям.

Якщо розглядати задачі комбінаторної оптимізації, то при накладанні певних умов вони мають дискретно-неперервну структуру. Це обумовило можливість використання як континуальних, так і дискретних моделей зазначених задач. При проектуванні вхідних даних комбінаторної задачі при відображенні *P*-простору вхідної інформації *G* на простір результуючої інформації G^*, виникає етап неперервної оптимізації, обумовлений релаксацією задачі.

Розглянемо задачі, моделями яких є екстремальні задачі оптимізації на перестановках, розміщеннях, сполученнях та ін.

Основна увага при цьому приділяється дослідженню фундаментальних властивостей зазначених множин та розгляд елементів таких множин як вершин деяких графів. Ці властивості дозволяють застосовувати відомі підходи до розв'язання задач комбінаторної оптимізації та розвинути нові. Розглянемо властивості типових видів комбінаторних конфігурацій.

1.2.1. Властивості перестановок

Практика показала, що найчастіше комбінаторними об'єктами є перестановки. Зокрема, перестановкою із n елементів називається така впорядкована множина з n елементів, які різняться між собою порядком їх розміщення. Надалі, якщо не оговорюється протилежне, будемо розглядати комбінаторні множини, породжені дійсними числами.

Розглянемо необхідні поняття, що визначають комбінаторні множини.

Вимірність підпростору $L \subset R^k : dim L$. Опуклу оболонку множини M позначимо $conv\, M$.

Нехай задана мультимножина $A = \{a_1, a_2, \ldots, a_q\}$, її основа $S(A) = \{e_1, e_2, \ldots, e_k\}$, тобто набір її різних елементів, де $e_j \in R^1 \ \forall j \in N_k$ і кратність елементів $k(e_j) = \alpha_j$, $j \in N_k$, $\alpha_1 + \alpha_2 + \ldots + \alpha_k = q$. Мультимножина B з основою $S(B)$ називається підмультимножиною мультимножини A з основою $S(A)$, якщо $S(B) \subset S(A)$ і для кожного елемента $a \in S(B)$ виконується нерівність $k_B(a) \le k_A(a)$.

Впорядкованої n-вибіркою з мультимножини A називається набір:

$$a = \left(a_{i_1}, a_{i_2}, \ldots, a_{i_n}\right), \tag{1.3}$$

де $a_{i_j} \in A \ \ \forall i_j \in N_n$, $\forall j \in N_n$, $i_s \ne i_t$, якщо $s \ne t \ \forall s \in N_n$, $\forall t \in N_n$.

Множина перестановок з повтореннями з n дійсних чисел, серед яких k різних, називається загальною множиною перестановок і позначається $P_{nk}(A)$. Це множина упорядкованих n-вибірок вигляду (1.3) з мультимножини A за умови $n = q > k$.

При $n=k=q$ маємо множину перестановок без повторень. Позначимо її P_n. Очевидно, що $P_n(A)=P_{nn}(A)$. У тих випадках, коли конкретно не будемо вказувати вид множини перестановок, будемо записувати ці множини як P_n. Елементами множини перестановок P_n є упорядковані набори $a=(a_1,a_2,\ldots,a_n)$ із n символів. Кожен символ a_i входить у набір тільки один раз і є представником множини $A_i, i\in N_n$.

Елементами множини перестановок з повтореннями P_n є упорядковані набори, що складаються із n символів, причому кожен символ $a_i \in A_i, i\in N_k, k\le n$ входить у набір n_i раз, а $\sum_{i=1}^{k} n_i = n$, тоді розглядається множина перестановок з повтореннями.

Якщо розглядати множину P_{nk}, то зрозуміло, що всяка точка $a=\left(a_{j_1},a_{j_2},\ldots,a_{j_n}\right)\in P_{nk}$ має таку властивість, що її координати приймають α_1 значень a_1, α_2 значень a_2 і т. д., α_k значень a_k.

Слід зазначити, що опуклою оболонкою загальної множини перестановок є загальний переставний многогранник, властивості якого описано в роботах [87, 245].

Далі розглянемо властивості розміщень та сполучень.

1.2.2. Властивості розміщень та сполучень

Розглянемо множину k-розміщень без повторення, тобто A − множина, $q=k$, $A=S(A)$. За такої умови множину всіх упорядкованих n-вибірок з мультимножини A вигляду (1.2) називають множиною n-розміщень без повторення з k різних дійсних чисел множини A. Позначимо цю множину розміщень $A_k^n(A)=A_q^n$. Якщо $n=k$, то множина A_q^n є множиною $P_n(A)$ перестановок n-різних дійсних чисел, що складають A.

Розглянемо множину q-розміщень з повтореннями з k різних дійсних чисел з мультимножини A, в якій q елементів.

Розглянемо загальну множину n-розміщень.

Нехай $A=\{a_1,a_2,\ldots,a_q\}$ як і раніше – мультимножина з основою $S(G)=(e_1,e_2,\ldots,e_k)$ та первинною специфікацію $[A]=(q_1,q_2,\ldots,q_k)$, де $q_i\le k\forall i\in N_k$. Сукупність усіх упорядкованих *n*-вибірок вигляду (1.3) з мультимножини *A* будемо називати загальною множиною *n*-розміщень і позначити її $A_{qk}^n(A)=A_{qk}^n$. Зазначимо, що оскільки елементи первинної специфікації мультимножини *A* задовольняють умови $q_i\le k$, то в кожному елементі множини A_{qk}^n не більше q_i елементів $e_i\in S(A)\,\forall i\in k$. Якщо ж $q_i<k\ \forall i\in k$, то, очевидно, що в кожному елементі загальної множини розміщень $A_{qk}^n(A)$ немає однакових чисел e_i, $i\in N_k$ (на відміну від множини $A_{qk}^n(A)$, де є k елементів, що складаються з однакових чисел e_i, $i\in N_k$). Зазначимо також, що при $k=q$, тобто при $q_i=1\quad\forall i\in N_k$, множина $A_{qk}^n(A)$ збігається з множиною $A_k^n(A)=A$ розміщень без повторень: $A_{kk}^n(A)=A_k^n(A)$, а при $q_i=n\ \forall i\in N_k$, тобто при $q=nk$, множина $A_{qk}^n(A)$ перетворюється на множину $A_k(A)$ розміщень з повтореннями: $A_{qk}^n(A)=A_k^n(A)$.

Якщо $n=q$, то множина $A_{qk}^n(A)$ є загальною множиною перестановок $P_{nk}(A)$ з елементів мультимножини *A*.

Як відомо [49, 51, 169], опуклою оболонкою точок загальної комбінаторної множини розміщень є загальний многогранник розміщень $M_{qk}^n(A)=conv\,A_{qk}^n(A)$. Загальний многогранник розміщень $M_{qk}^n(A)$ можна представити сукупністю всіх розв'язків такої системи нерівностей:

$$\sum_{j=1}^{|\omega|}a_{q-j+1}\le\sum_{i\in\omega}x_i\le\sum_{j=1}^{|\omega|}a_j,\forall\omega\subset N_n.$$

Визначення 1.3. Вектор *x* є вершиною загального многогранника розміщень $M_{qk}^n(A)$ тоді і тільки тоді, коли він представляє собою перестановку чисел, $a_1,\ldots,a_s,\ldots,a_{q-r+1},\ldots,a_q$, де $s\in N_n, r\in\{N_n\cup 0\}$, $s+r=n$.

Розглянемо сполучення. Множина всіх *k*-вибірок з множини $S(A)$ вигляду $\left(a_{i_1},a_{i_2},\ldots,a_{i_k}\right)$ називають множиною *k*-сполучень без повторення з n різних дійсних чисел, якщо виконується умова $e_{i_1}<e_{i_2}<\ldots<e_{i_k}$. Позначимо цю множину сполучень $S_n^k(A)$. Сукупність всіх *k*-вибірок вигляду (1.3) з мультимножини A називається загальною множиною сполучень $S_{qn}^k(A)$, якщо виконується умова $a_{i1}\le a_{i2}\le\ldots\le a_{ik}$.

1.2.3. Властивості розбиття

При розв'язанні комбінаторних задач часто використовується такі добре відомі конструкції з елементів скінченної множини, як сполучення, розміщення, перестановки і т. д. Але поряд з добре відомими комбінаторними конфігураціями виділяються складніші конструкції – множини розбиття. Інтерес до таких множин обумовлений різними прикладними задачами, які добре описуються за їх допомогою. У літературі розглядається розбиття множини і розбиття числа. Розглянемо дані поняття більш докладніше.

Сукупність $A_1,A_2,\ldots,A_k$ не порожніх попарно диз'юнктивних підмножин множини *M* називається його розбиттям, якщо множина *M* рівна їх об'єднанню:

$$M=A_1\cup A_2\cup\ldots\cup A_k, A_i\cap A_j=\varnothing, i\ne j.$$

Розбиття $A=\{A_1,A_2,\ldots,A_k\}$ множини *M* називається дробленням розбиття $B=\{B_1,B_2,\ldots,B_l\}$ тієї ж множини, якщо кожна із множин A_i, повністю міститься в одній із множин B_j, тобто для кожного $i\in N_k$ знайдеться єдине $j\in N_l$, таке що $A_i\in B_j$. Будемо говорити, що *B* складніше (крупніше), ніж *A*.

Розглядаються впорядковані і невпорядковані розбиття. Розглянемо впорядковані розбиття множини. Визначимо число розбиттів скінченої множини *S*, де $|S|=n$, на *k* різних підмножин $S = S_1 \cup S_2 \cup \ldots \cup S_k$, що попарно не перетинаються, $|S| = n_i$, $i = 1, 2, \ldots, k$ і $\sum_{i=1}^{k} n_i = n$. Послідовність різних множин $S_1, S_2, \ldots, S_k$ розглядається як упорядкована послідовність підмножин. При формуванні упорядкувань $S_1, S_2, \ldots, S_k$ послідовностей на перше місце підмножину S_1 можна вибрати $C_n^{n_1}$ способами, на друге місце підмножину S_2 можна вибрати $C_{n-n_1}^{n_2}$ способами із $n - n_1$ елементів, що залишилися і т. д., на останнє місце множину S_k можна вибрати $C_{n-n_1-n_2-\ldots-n_{k-1}}^{n_k}$ способами із $n - n_1 - n_2 - \ldots - n_{k-1}$ елементів, що залишилися. За правилом прямого добутку отримуємо, що загальне число впорядкованих розбиттів множини S на k підмножин рівно

$$C_n^{n_1} C_{n-n_1}^{n_2} \ldots C_{n-n_1-\ldots-n_{k-1}}^{n_k} = \frac{n!}{n_1! n_2! \ldots n_k!},$$

що співпадає з числом $P(n_1, n_2, \ldots, n_k)$ перестановок з повтореннями.

Зауваження. Встановимо взаємоднозначну відповідність між впорядкованим розбиттям множини і перестановками з повтореннями. Кожній перестановці з повтореннями можна поставити у відповідність впорядковане розбиття множини номерів елементів $S = \{1, 2, \ldots, n\}$ у перестановці на підмножини $S_1, S_2, \ldots, S_k$ де S_i – множина номерів елементів *i*-го типу в перестановці. Очевидно, що дана відповідність між перестановками з повтореннями і розбиттям є взаємнооднозначною.

Загальна кількість елементів невпорядкованого розбиття буде в $m_1! m_2! \ldots m_n!$. раз менше, ніж впорядковане. Отже

$$N(m_1, m_2, \ldots, m_n) = \frac{n!}{(1!)^{m_1} (2!)^{m_2} \ldots (n!)^{m_n} m_1! m_2! \ldots m_n!}.$$

Зазначимо, що якщо виконано впорядковане розбиття числа на n підмножини різної довжини (потужності), то вони співпадають з невпорядкованим розбиттям. В цьому випадку всі $m_i \notin \{0,1\}$.

Формула

$$\left(x_1 + x_2 + \ldots + x_k\right)^n = \sum_{n_1+n_2+\ldots+n_{k=n}} \frac{n!}{n_1!n_2!\ldots n_k!} x_1^{n_1} x_2^{n_2} \ldots x_k^{n_k}$$

називається поліноміальною, де додавання виконується по всім розв'язкам рівняння $n_1 + n_2 + \ldots n_k = n$ у цілих невід'ємних числах $n_i \geq 0,\ i = 1, 2, \ldots, k$.

Розбиття часто виникає при формуванні повних прообразів значень функції: якщо взяти деяке відображення $f : M \rightarrow R$ і співставити кожному $r \in R$ множину $A_r = \breve{f}\left(r\right) = \left\{i \in M \mid f\left(i\right) = r\right\}$, то набір непорожніх множин A_r матиме всі властивості розбиття. Позначимо це розбиття через A_f.

Якщо задано три множини $M,\ R$ і S і відображення $f : M \rightarrow R$ і $g : R \rightarrow S,$ то суперпозиція даних відображень, тобто відображення $gf : M \rightarrow S,$ де $\left(gf\right)\left(i\right) = g\left(f\left(i\right)\right)$ задає розбиття A_{gf} яке складніше, ніж розбиття A_f.

Тепер візьмемо два розбиття множини $M : A = \left\{A_1, A_2, \ldots, A_k\right\}$ і $B = \left\{B_1, B_2, \ldots, B_l\right\}$. Розбиття $C = \left\{C_1, C_2, \ldots, C_m\right\}$ називається *добутком розбиттів* A і B і позначається $C = A \cdot B,$ якщо воно являється подрібненням обох даних розбиттів, при чому найбільшим із всіх подрібнень.

На практиці часто доводиться розв'язувати задачі, які вимагають операції з натуральними числами і їх сумами. Зручним комбінаторним трактуванням для таких задач виявилося поняття розбиття числа.

Слід зазначити що існує добуток двох розбиттів.

Розглянемо зв'язок розбиттів і прямих добутків. Нехай задано дві множини: A з розбиттям A_i і B з розбиттям B_j. На їх добутку $C = A \times B$ легко визначається розбиття C, що складається із добутків $A_i \times B_j, A_i \in A, B_j \in B,$ яке можна назвати добутком цих розбиттів. Дійсно, розбиття A і B легко перетворюються в розбиття C_A і C_B множини C, тоді розбиття C являється добутком даних розбиттів: $C = C_A \cdot C_B$.

Визначення 1.4. Розбиттям натурального числа n є його представлення невпорядкованою сумою натуральних доданків: $n = n_1 + ... + n_r$, де доданки n_i називаються частинами, а їх число r – рангом розбиття.

Композиція – це представлення натурального числа n впорядкованою сумою натуральних доданків. Таким чином, композиції можна розглядати як «впорядковане розбиття».

В загальному вигляді екстремальні задачі на розбиттях можуть бути сформульовані у вигляді питання: наскільки багато існує розбиттів, що задаються в даній відповідності? Вибір конкретної відповідності між розбиттям визначається умовами практичної задачі, саме тієї, для якої розбиття з такою відповідністю служить комбінаторною схемою.

Далі будемо розглядати задачі, що моделюються екстремальними задачами на вище зазначених комбінаторних конфігураціях.

Висновки до розділу

При розв'язанні комбінаторних задач часто використовуються такі добре відомі конструкції з елементів скінченої множини, як сполучення, розміщення, перестановки й т. п. В розділі визначаються поняття цих конструкцій через формалізацію комбінаторних схем з використанням поняття відображення однієї скінченої множини *X* в іншу *Y*. На основі цього підходу побудовані комбінаторні конфігурації, які відповідають найпростішим комбінаторним об'єктам: розміщенням, перестановкам, комбінаціям, розбиттям та іншим об'єктам, а також розглянуті їх властивості, які в подальшому використовуються при розв'язуванні задач.

РОЗДІЛ 2. СТАН ПРОБЛЕМИ, НАПРЯМКИ ДОСЛІДЖЕНЬ ТА МЕТОДИ РОЗВ'ЯЗАННЯ ЕКСТРЕМАЛЬНИХ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ

Екстремальні задачі дискретного характеру на комбінаторних конфігураціях постійно зустрічаються при розв'язуванні практичних задач. Проте довгий час вони не привертали до себе уваги, оскільки в більшості випадків для їх розв'язання знаходився деякий алгоритм розв'язування, зазвичай типу перебору. Широке застосування комп'ютерів істотно змінило положення, так як стало можливим розв'язувати екстремальні задачі великої розмірності. Виявилось, що для таких задач різні удосконалення природних алгоритмів можуть давати істотний виграш в часі роботи або в потребах необхідної пам'яті. Необхідність практичного розв'язання широкого кола екстремальних задач на комбінаторних конфігураціях привела до появи великої кількості методів, що ґрунтуються на природних алгоритмах, а у багатьох випадках і до побудови принципово нових методів, що враховують специфіку задачі. Це в свою чергу привело до необхідності порівняння якості теоретичних методів, а також дослідження загальних принципів побудови кращих комбінаторних алгоритмів.

2.1. Математична постановка задачі комбінаторної оптимізації та її властивості

У загальному випадку задачу оптимізації можна представити кортежем $< f, X, \Pi, D, ext >$, де $f : X \to R^1$ – задана цільова функція задачі, R^1 – числова пряма, X – простір розв'язків задачі, Π – предикат, який визначає підмножину $D \subseteq X$ допустимих варіантів розв'язку згідно наявних обмежуючих умов, $ext \in \{min, max\}$ – напрям оптимізації. У цих позначеннях задачу оптимізації можна переписати у наступному вигляді: необхідно знайти $x^* \in D \subseteq X$ таке, що

$$x^* = arg \min_{x \in D \subseteq X} f(x). \qquad (2.1)$$

В деяких роботах під задачею комбінаторної оптимізації розуміється проблема пошуку екстремумів заданої цільової функції вигляду (2.1), коли X – комбінаторний простір. Під комбінаторним простором тут розуміється сукупність комбінаторних об'єктів певного типу, утворених із елементів заданої скінченної множини потужності n. Водночас, поняття «комбінаторний об'єкт» вживається як сполучення, перестановки, розміщення та ін. В зарубіжній літературі переважно вживається визначення задачі комбінаторної оптимізації, де простір варіантів X вважається скінченною множиною. Таке тлумачення не дозволяє строго формально окреслювати окремі класи задач комбінаторної оптимізації, такі, наприклад, як дискретне, цілочисельне чи булеве програмування. Більш того, воно часто призводить до фактичного ототожнення понять дискретної та комбінаторної оптимізації, яке і спостерігається в ряді робіт.

Найбільш вживане формулювання екстремальних комбінаторних задач можна визначити наступним чином: є n-множина елементів, на ній визначається скінченна множина комбінаторних конфігурацій $A=\{\pi_i\}=\{a_1,a_2,\ldots,a_q\}$, де $q>n$. Під комбінаторними конфігураціями $\pi_i=a_1,a_2,\ldots,a_q$ можна розуміти перестановки, розміщення, сполучення, розбиття, різні послідовності й т. п. На множині A задається функція $f(x)$. Потрібно відшукати екстремум $f(x)$ (максимум або мінімум) і елементи множини A, які цей екстремум доставляють.

Отже, задана деяка скінченна дискретна множина A. Комбінаторним простором $X=X_A$ назвемо сукупність усіх комбінаторних об'єктів визначеного вигляду з елементів множини A.

Тоді задачу (2.1) в загальному вигляді слід розуміти так: визначити точку x^* з простору X, що доставляє екстремум деякому функціоналу $f(x)$, тобто

$$f(x^*)=\underset{x\in X}{extr}\ f(x)$$

і задовольняє заданим умовам.

Саме формулювання екстремальних комбінаторних задач диктує широкий вибір операцій, які застосовуються для їх розв'язання. По-перше, треба вміти робити генерацію множини елементів A і мати у своєму розпорядженні відповідну множину значень функції $f(x)$. По-друге, треба розробити методику порівняння цих значень і виділення з них максимального або мінімального. Операція переліку практично рідко виявляється здійсненною, тому що кількість можливих комбінацій може бути занадто великою. Труднощі, пов'язані з переліком варіантів і порівнянням значень, досить значні. Саме вони й були перешкодою для розвитку деяких розділів комбінаторного аналізу, незважаючи на їх очевидну актуальність. Лише застосування в математичній практиці обчислювальних машин створило можливості для розв'язання низки екстремальних задач.

Як відомо, більшість оптимізаційних задач комбінаторного типу можуть бути зведені до задач цілочисельного програмування. У роботі [203] встановлені деякі достатні умови зведення комбінаторних задач до задач цілочисельного лінійного програмування та показано, що не існує загального алгоритму, який знаходить таке зведення, навіть якщо воно існує. Але, зведення оптимізаційної комбінаторної задачі до вигляду оптимізаційної цілочисельної задачі не завжди доцільно, бо при цьому втрачаються комбінаторні властивості задачі та їх адекватний зміст [203, 206, 207].

Загальна схема зв'язку екстремальних комбінаторних задач з методами лінійного програмування виглядає приблизно так: елементи π_i інтерпретують, як точки евклідового простору, щоб «цільова» функція $f(x)$ стала лінійною формою. Розглядається задача знаходження екстремуму цієї функції на опуклій оболонці заданих точок (тобто на опуклому многограннику). Справді, екстремум лінійної форми на многограннику досягається в одній з вершин, які входять у множину розглянутих елементів. Задача ж знаходження екстремуму лінійної форми і є задачею лінійного програмування. Особливістю комбінаторних задач при такому зведенні залишиться те, що при знаходженні розв'язку варто обмежуватися лише точками із цілочисловими координатами.

Тоді розв'язок комбінаторних екстремальних задач являє собою набір $(a_1, a_2, \ldots, a_n)$ чисел $1, 2, \ldots, n,$ множина яких представляє собою комбінаторну конфігурацію. Необхідно знайти значення функції $f(x)=\sum_{i=1}^{n} c_i x_i$ на скінченній підмножині вибраних елементів конфігурацій.

Зазначимо, що коефіцієнти $c_1, c_2, \ldots, c_n$ цільової функції $f(x)$ як правило є невпорядкованими, але їх завжди можна впорядкувати за зростанням чи спаданням. Призначимо кожному коефіцієнту відповідний індекс з множини натуральних чисел $N_m=\{1,2,\ldots,m\}$, тоді для деякої множини коефіцієнтів $C=(c_1, c_2, \ldots, c_n)$ розглянемо дворядковий запис відображення $\varphi: N \to C$:

$$\varphi=\begin{pmatrix} 1 & 2 & \ldots & m \\ \varphi(1) & \varphi(2) & \ldots & \varphi(m) \end{pmatrix}, \tag{2.2}$$

де $\varphi(i)\in C, i \in N_m$. Упорядкований нижній рядок запису визначає коефіцієнт цільової функції і його місце в початковому векторі *C*. На відміну від елементів множини N_m, значення координат можуть співпадати і порядок розміщення різних значень координат суттєвий. Для подальшого викладу матеріалу введемо наступне

Визначення 2.2. Нормалізацією функції $f(x)=\sum_{i=1}^{n} c_i x_i$ назвемо операцію відображення перестановки $\varphi: N \to C$, що встановлює впорядкування коефіцієнтів $c_1, c_2, \ldots, c_n$ цільової функції за зростанням (спаданням).

Тоді функція є нормалізованою, якщо її коефіцієнти є впорядкованими. Зрозуміло, що для будь-якої функції $f(x)$ можна встановити порядок коефіцієнтів, тобто нормалізувати функцію. Щоб повернутися до попередньої початкової форми лінійної функції, необхідно зробити обернене відображення: $\varphi^{-1}: C \to N$.

Слід зазначити, що розв'язок, який одержується при зведенні і розв'язанні нової задачі методами цілочисельного лінійного програмування без врахування специфіки комбінаторної задачі є не завжди адекватним до постановки вихідної задачі. Хоча існує ряд алгоритмів, що враховують для цілочисельної моделі задачі її комбінаторні властивості [52, 58–60, 98, 125, 216–221, 242, 243]. Таким чином, не існує чітко вираженої границі між оптимізаційними задачами, що представляються цілочисельними і комбінаторними моделями. Тому дуже важливо для розв'язання задачі підібрати відповідну модель задачі, що враховувала б найважливіші властивості й особливості задачі так, щоб врахування цих властивостей допомогло побудувати ефективні методи розв'язання задачі.

В іншому вигляді задачі визначені на просторі $X = X_A$ комбінаторних об'єктів, породжуваних деякою дискретною скінченною множиною *A*, можна інтерпретувати так: екстремум функціоналу $f(x)$, заданого на просторі *X*, визначається на всьому чи частині цього простору *R*, тобто $R \subseteq X$. Таким чином, оптимізаційну комбінаторну задачу в загальному вигляді можна записати таким способом: визначити $x^* \in X_A$ із співвідношення

$$f\left(x^*\right) = \underset{x \in R \subseteq X}{extr}\, f\left(x\right)$$

або у вигляді

$$x^* = arg\left(\underset{x \in R \subseteq X}{extr}\, f\left(x\right)\right).$$

Зрозуміло, що в силу скінченності множини *A*, оптимізаційна комбінаторна задача має розв'язок завжди, і цей розв'язок найчастіше неєдиний.

Таким чином, відпадає питання про дослідження задачі на можливість розв'язання (у випадку нескінченності множини *A* це питання потребує подальшого дослідження).

Однак проблема приймає форму практичної можливості розв'язання, тобто того, чи не перевищують запити даної оптимізаційної комбінаторної задачі наявні ресурси (швидкодія, пам'ять комп'ютера, потужність мікропроцесору та ін.). Отже, питання про практичну можливість розв'язання задачі залишається одним з основних питань обчислювальної математики.

Багато вчених в своїх працях [37, 58–60, 150–153] підкреслюють важливість комбінаторного представлення визначених класів дискретних оптимізаційних задач в різних інтерпретаціях, що дає можливість більш адекватно описати постановку прикладної моделі.

З погляду теорії графів, багато комбінаторних задач оптимізації мають наступне формулювання: відшукати серед деякої множини шляхів L мінімальний (або максимальний), тобто шлях, що володіє мінімальним (або максимальним) значенням λ. Як множина L може бути обрано, наприклад, множина всіх гамільтонових шляхів.

Слід зазначити, що за допомогою теорії графів добре описуються багато типів комбінаторних задач. При цьому графічні подання є не просто ілюстраціями, але й дозволяють одержувати нові підходи до їх розв'язання, а також результати.

Формальні постановки оптимізаційних комбінаторних задач можна розділити на два класи: безумовні та умовні. Ці поняття аналогічні відповідним поняттям з недискретної оптимізації. Умовність задачі найчастіше залежить від простору, у якому вона формулюється. Як правило, введення додаткових обмежень ускладнює задачу в алгоритмічному відношенні, однак зменшує область допустимих розв'язків, на якій розглядається задача.

Розглянемо надалі необхідні поняття комбінаторних конфігурацій, на яких розглядаються екстремальні задачі та їх властивості.

2.2. Моделі задач комбінаторної оптимізації

Новим теоретичним підходом для розв'язування задач проектування автоматизованих систем управління та інформаційних технологій є застосування методів комбінаторного аналізу, а саме, тематики екстремальних комбінаторних задач на різних комбінаторних конфігураціях. Високий ступінь абстракції постановок і розв'язання екстремальних комбінаторних задач дозволяє використовувати їх при проектуванні і технічних, і програмних засобів автоматизованих систем управління та при розв'язуванні різних прикладних задач економіки, фінансів, статистики та ін. Сукупність комбінаторних об'єктів утворює комбінаторні моделі, які на основі апріорної інформації про розв'язування прикладної задачі забезпечують повний опис всіх їх показників та характеристик. Розглянемо огляд прикладних задач з [231, 232], що моделюються екстремальними задачами на комбінаторних конфігураціях. Зокрема, в даному пункті представлені як загально відомі дискретні оптимізаційні задачі, так і формалізовані порівняно недавно. Відомою класичною задачею на комбінаторній множині є задача про ранець, моделлю якої є цілочисельна лінійна модель. Також до загальновідомих задач комбінаторної оптимізації відноситься задача комівояжера. Як показує практика, найбільш вдалим для її розв’язання є алгоритм, що враховує її комбінаторні властивості.

2.2.1. Задача інтерполяції дискретної функції

Нехай на множині точок $A=\{a_1, a_2, \dots, a_n\}$ експериментальним шляхом отримані значення деякої функції $f(a_1), f(a_2), \dots, f(a_n)$. Треба визначити таку підмножину точок $x \subset A$, що містить не більше N елементів, $N < n$, щоб інтерполяція за цими точками функції $f(a)$ функцією $\psi(a)$, $a \in [a_1, a_n]$, визначеного вигляду давала мінімальне відхилення

$$\varphi(x) = \sum_{i=1}^{n} | f(a_i) - \psi(a_i) |,$$

тобто треба визначити $x^* \in M_A$, де M_A – множина, породжена елементами дискретної скінченної множини A, для якого

$$\varphi(x^*) = \min_{x \in R} \varphi(x),$$

де область R описана обмеженням $x^* \leq N$.

2.2.2. Задача мінімізації часу розв’язування

Часто при розв’язанні задач в автоматизованих системах управління виявляється істотною черговість, у якій вони надходять для розв’язання в оперативну пам’ять обчислювальної системи; від цього може істотно залежати, наприклад, час розв’язання всіх задач множини A і низка інших показників ефективності організації обчислювальних процесів у системі. Останнє може виявитися важливим, наприклад, якщо задачі взаємозалежні між собою, або коли розв'язання задач множини A здійснюється на великих схемах мультипрограмуванням. До такого роду задач комбінаторного типу зводяться також багато задач календарного планування (наприклад, коли розраховується черговість запуску у виробництво деталей, оброблюваних на декількох різнотипних верстатах відповідно до деякої технологічної програми). У таких випадках задачу доцільно формулювати в метричному просторі S_A розміщень.

Зокрема, нехай задача полягає в наступному. З даної множини A задач виділити підмножину потужності не меншу D й упорядкувати її так, щоб час розв'язання її на комп'ютері був мінімальним. Природно тут точку розв'язання задачі шукати серед елементів простору S_A.

2.2.3. Задача оптимізації ресурсів комп'ютера

Нехай задана множина $A=\{a_1,a_2,\ldots,a_n\}$ задач, що підлягають розв'язанню на комп'ютері. Функції $v=v(x)$ і $t=t(x)$ виражають відповідно пам'ять комп'ютера, необхідну для розв'язання підмножини задач $x\subset A,$ і сумарний час їх розв'язання. Тоді вище зазначені функції виражаються залежностями

$$v(x)=\sum_{a_i\in x}v(a_i),\ t(x)=\sum_{a_i\in x}t(a_i),$$

які відповідно необхідно мінімізувати, тобто визначити $x^*\in M_A$, що задовольняє умовам $v(x)\le V,\ t(x)\le T$ і доставляє максимум функціоналу

$$f(x)=\sum_{a_i\in x}\varphi(a_i),$$

де $\varphi(a_i)=\alpha_1 v(a_i)+\alpha_2 t(a_i),\ V, T,\ \alpha_1,\ \alpha_2$ – сталі.

2.2.4. Задача теорії розкладів

Одна із задач теорії розкладів для m машин полягає в наступному. Кожна з деталей a_i, що належать множині $A=\{a_i\}_{i=1}^n$, повинна послідовно пройти обробку на машинах $B_1,\ldots,B_m$.

Потрібно знайти x^*, що мінімізує деяку функцію $\Phi(x)$ за умови, що $T(x) \leq T$, де $T = const$, $T(x)$ – час виконання плану x на машинах $B_1, B_2, \ldots, B_m$ відповідно до заданих програм обробки деталей.

Формальна постановка даної задачі може представлятися у двох просторах. Якщо не відомо, чи є величина T достатньою для того, щоб у шуканий план увійшли всі елементи множини A, то доцільно задачу розглядати в просторі розміщень S_A, тобто знаходити $x^* \in S_A$, мінімізуючи функцію $\Phi(x)$ за умови $T(x) \leq T$.

Якщо ж відомо, що обмеження $T(x) \leq T$ допускає розклад, що включає всі елементи множини A, то доцільно представляти задачу в просторі перестановок P_A. Таким чином, наявність додаткової інформації про потужності x^* дозволяє використати у формулюванні задачі простір меншої розмірності $P_A^* < S_A^*$, що позитивно позначається, загалом кажучи, на часі розв'язування задачі.

Зазначимо, що для практичного використання цієї постановки задачі необхідно ввести уточнення, що стосуються способу обчислення функцій $\Phi(x)$ і $T(x)$ відповідно до заданих програм обробки кожної з деталей $a_i \in A$, а також додатково ввести деякі інші обмеження типу: (на одній машині обробляється одна деталь та ін.).

2.2.5. Задача оптимізації оргструктури

Розглядається множина підприємств, зв'язки між якими фіксовані. Під зв'язками розуміються основні укрупнені показники, що характеризують матеріально-речовинні, фінансові й інформаційні точки. Ставиться задача оптимізації триступінчастої структури керування даною множиною підприємств.

Отже, задана сукупність $A = \{a_i\}_{i=1}^n$ підприємств і сукупність $B = \{a_i, a_j\}_{i,j=1}^n$ зв'язків між ними, $i \neq j$. Кожному підприємству a_i відповідає число $p(a_i)$, що вимірює обсяг виробництва.

Кожному зв’язку $\left(a_i, a_j\right)$відповідає число $f\left(a_i, a_j\right)$, що вимірює інтенсивність цього зв’язку. Необхідно знайти розбиття сукупності A на підмножини $x_1^*, x_2^*, \ldots, x_k^*$; кількість їх заздалегідь не фіксована й мінімізується сума взаємодій підмножин $x_1^*, x_2^*, \ldots, x_k^*$ при деяких додаткових умовах.

Цю задачу природно сформулювати в просторі розбиттів N_A. Нехай

$$X \in N_A \text{ і } X = \left\{x_1^*, x_2^*, \ldots, x_k^*\right\}.$$

Тоді сума взаємодій підмножин $x_1, x_2, \ldots, x_k$ визначиться таким способом:

$$F(X) = \sum_{r=1}^{k} \sum_{s=1}^{k} \sum_{a_i \in x_r} \sum_{a_j \in x_s} f\left(a_i, a_j\right).$$

Нехай, далі, c – обмеження на обсяг виробництва в одній підмножині; $\Delta = \left\|\delta\left(a_i, a_j\right)\right\|_{i,j=1}^{n}$ – матриця заборон,

$$\delta\left(a_i, a_j\right) = \begin{cases} 1, \textit{ якщо } a_i, a_j \textit{ допускається,} \\ \textit{розміщати в одне об'єднання,} \\ 0 - \textit{ в протилежному випадку.} \end{cases}$$

k – максимальне кількість підприємств, що дозволяється вміщати в одне об’єднання.

Таким чином, задача полягає в наступному.

Знайти елемент X^* простору розбиттів N_A, мінімізуючи функцію $F(X)$ на допустимій області $R \subset N_A$, тобто

$$F\left(X^*\right) = \min_{x \in R \subset N_A} F(X),$$

де елементи допустимої області R задовольняють перерахованим вище вимогам, а саме: якщо $X \in R \subset N_A$, $X = \left\{x_1, x_2, \ldots, x_k\right\}$, то для $s = 1, 2, \ldots, k$ маємо а) $\sum\limits_{a_i \in x_s} p\left(a_i\right) \le c$; б) для всіх $a_i, a_j \in x_s$ $\delta\left(a_i, a_j\right) = 1$; в) $x_s \le k$.

2.2.6. Екстремальні задачі на розбиттях

А) Задача розбиття множини задач. Нехай, $\vartheta(x)$ позначає об'єм пам'яті комп'ютера, необхідний для розміщення програм задач із множини x.

Задача полягає в тому, щоб розбити задану множину задач на мінімальну кількість підмножин, що взаємно не

перетинаються, таким чином, щоб пам'ять комп'ютера, зайнята кожною із цих підмножин, не перевищувала заданого числа *V*. Аналогічні задачі розглядалися в роботі [155].

Комбінаторна модель цієї задачі представляється таким чином.

Знайти елемент X^* простору розбиттів N_A, що задовольняє умовам

$$\vartheta\left(x_i^*\right) \le V, \quad x_i^* \in X^*, \quad i = 1, 2, \ldots, X^*,$$

і такий, щоб $X^* = \min\limits_{X \in N_A} X$.

В) Задача дослідження процесу розподілу пам'яті комп'ютера.

Задачі на розбиттях чисел довгий час не виділялися як екстремальні, але в зв'язку з розвитком інформаційних технологій та необхідністю проектування автоматизованих систем управління для подолання проблем в цих галузях виникла потреба в розв'язуванні екстремальних комбінаторних задач на розбиттях.

Розглянемо одну з комбінаторних моделей, що дозволяють досліджувати процес розподілу оперативної пам'яті комп'ютера з сегментною організацією програм і даних [11].

Відомо, що у будь-який момент часу функціонування автоматизованих систем управління відбувається вплив зовнішньої фрагментації на процес розподілу пам'яті, і це достатньо повно характеризують наступні параметри:

- кількість вільних (зайнятих) ділянок пам'яті;
- розмір вільних (зайнятих) ділянок;
- сумарний розмір вільної (зайнятої) пам'яті.

Запити на виділення пам'яті в цих дослідженнях у будь-який момент часу функціонування автоматизованих систем управління достатньо повно характеризуються наступними параметрами:

– кількістю запитів в черзі на виділення пам'яті;

– необхідними розмірами неперервних ділянок адресного простору пам'яті або розмірами запитів;

– сумарним розміром пам'яті, потрібної для задоволення запитів в черзі.

Нехай Q – розмір оперативної пам'яті комп'ютера автоматизованої системи управління, а N – сумарний розмір вільної пам'яті, який в процесі функціонування системи приймає значення $N \in Z^+$, $N \le Q$, де Z^+ – множина цілих додатних чисел. Через вплив зовнішньої фрагментації пам'ять розміром N виявиться подрібненою на r вільних фрагментів, представлених ділянками неперервного адресного простору пам'яті. Такий стан вільної пам'яті можна інтерпретувати як вектор

$$z(N)(n_1,\dots,n_r), N = \sum_{i=1}^{r} n_i,\ n_1 \ge n_2 \ge \dots \ge n_r;$$

$N \in Z^+$ де n_i – розмір i-ї вільної ділянки пам'яті, а r – кількість таких ділянок.

При моделюванні запитів на виділення вільної пам'яті з $z(N) = (n_1, n_2, \dots, n_r)$ припускаємо, що вони можуть надходити або одночасно, тобто групами $q(K) = (k_1, k_2, \dots, k_t)$ або поодинці, де k_j – необхідний розмір вільної пам'яті для j-го запиту. Групу запитів також будемо тлумачити як вектор, тобто

$$q(K) = (k_1, k_2, \dots, k_t), \sum_{j=1}^{t} k_j = K, k_1 \ge k_2 \ge \dots \ge k_t,\ k_j \in Z^+.$$

Дві групи запитів вважатимемо різними, якщо вони різні як вектори.

Елементи n_i, d_l і k_j векторів $z(N)$, $g(D)$, $q(K)$ є натуральними числами. Отже, $z(N)$, $g(D)$, $q(K)$ можна тлумачити як розбиття чисел N, D, K відповідно, тобто $p(N) = (n_1, n_2, \dots, n_r)$, $p(K) = (k_1, k_2, \dots, k_t)$, $p(D) = (d_1, d_2, \dots, d_l)$, де частини розбиття n_i визначають розміри вільних ділянок адресного простору пам'яті, частини k_j – необхідні розміри неперервних ділянок адресного простору вільної пам'яті або розміри запитів на пам'ять, а частини d_m – розміри неперервних ділянок адресного простору зайнятої пам'яті. Ранги розбиття $p(N)$, $p(K)$, $p(D)$ визначають відповідно; r – число неперервних ділянок адресного простору вільної пам'яті, t – кількість запитів в черзі і l – число неперервних ділянок адресного простору зайнятої пам'яті.

Інтерпретація станів вільної і зайнятої пам'яті невпорядкованим розбиттям чисел дозволяє адекватно моделювати зовнішню фрагментацію пам'яті без врахування станів її адресного простору, що істотним чином спрощує проведення досліджень процесу розподілу пам'яті.

В задачі для подальших досліджень необхідно визначити: чи є в пам'яті вільні неперервні ділянки її адресного простору, необхідні для задоволення запитів, що надійшли в пам'ять? При такій постановці задачі не вимагається даних про стан адресного простору вільної пам'яті.

Представлення груп запитів на виділення пам'яті невпорядкованим розбиттям чисел також не протиставляється практичному сенсу досліджуваного процесу. Отже, розглянута модель є адекватним представленням як станів фрагментованої пам'яті, так і систем запитів на виділення пам'яті, які можуть утворюватися в процесі функціонування комп'ютера.

За допомогою множини розбиття чисел можна описати множину всіх можливих станів фрагментарної вільної пам'яті фіксованого розміру. Як вже наголошувалося, в процесі функціонування автоматизованої системи управління сумарний розмір вільної пам'яті комп'ютера змінюється в межах $0 \le N \le Q$, де Q − розмір пам'яті комп'ютера. Використовуючи наступну властивість множини розбиття чисел: $P(N_1) \cap P(N_2) \cap \ldots \cap P(N_r) = 0$, де $P(N_i)$ − множина розбиття числа N_i $\forall i \ne j, N_i \ne N_j$ можна показати, що множина станів фрагментарної вільної пам'яті комп'ютера розміром Q визначається множиною розбиття чисел

$$Z(Q)=\bigcup_{N=0}^{Q} Z(N)=\bigcup_{N=0}^{Q}\ \bigcup_{r=1}^{min(N,Q+1-N)} P_r(N).$$

Дані моделі досить наглядно представляють застосування екстремальних задач на розбиттях до розв’язування практичних проблем.

2.3. Сучасні методи розв’язання задач комбінаторної оптимізації

Методи розв’язування комбінаторних задач розвиваються досить інтенсивно. Очевидно, що швидко розповсюджуються методи, які краще і простіше враховують властивості і специфіку класів комбінаторних задач.

Загальні алгоритми для розв’язування екстремальних комбінаторних задач поділяються на дві категорії (в деякій мірі перетинні): алгоритми відтинаючої площини, що є розвитком симплекс-алгоритму, і перебірні алгоритми, засновані на розумному переборі всіх можливих розв’язків.

Основною ідеєю перебірних методів є перехід від повного перебору скінченної множини розв’язків до скороченого, направленого. Ці методи мають ряд позитивних якостей: гнучкість, універсальність, можливість застосування до різних задач комбінаторної оптимізації в будь-якій постановці. Найбільшого поширення набули методи комбінаторної оптимізації, що використовують методи гілок та меж [88], метод послідовного аналізу варіантів, методи побудови послідовності розв’язків, методи локальної оптимізації, метод динамічного програмування, апроксимаційно-комбінаторний метод [143] та ін. Евристичні алгоритми, що застосовуються при розв’язанні задач комбінаторної оптимізації, хоча досить швидко дають розв’язок, однак не гарантують його оптимальність.

2.3.1. Метод гілок та меж

Перші публікації про метод гілок і меж відносяться до початку 60-х років [230, 231]. У своїй роботі Ленд і Дойг метод гілок і меж застосований до оптимізаційних цілочисельних задач. Більш повно з врахуванням інших можливостей застосування і модифікацій описано метод гілок і меж в роботі Літтла, Мурті, Суїні, Керел [148, 149, 231] на прикладі задачі комівояжера. Тут же вперше вживається термін «метод гілок і меж» і повідомляється про успішний чисельний експеримент для задачі комівояжера з 40 містами.

Як відомо, задача комівояжера формулюється в просторі перестановок як задача на безумовний мінімум. Маємо $(n-1)!$ (потужність простору перестановок з n елементів) циклів, один або декілька з яких дають мінімальну довжину. Метод повного перебору тут явно непридатний. Метод гілок і меж дав результати, кращі в порівнянні з усіма відомими методами, що і зумовило його популярність.

Алгоритм Літтла може бути застосований і для асиметричних задач, які з'являються в різних додатках. Автори [250–254] наводять приклад з області календарного планування.

Припустимо, що протягом деякого періоду часу складальна лінія повинна збирати вироби різних пакетів. Вартість переходу від виробу типу i до типу j дорівнює c_{ij}. Яка послідовність типів збираних виробів мінімізує сумарні витрати? Це задача комівояжера, в якій немає потреби вважати $c_{ij} = c_{ji}$.

Схема алгоритму така: множина всіх можливих циклів розбивається на все менші і менші підмножини, для кожної з яких обчислюють мінімальні можливості витрат (довжину) циклу або їх нижні межі. Відповідно до обчислених мінімальних значень меж циклу проводять послідовне подібне розбиття підмножин і врешті-решт визначають оптимальний цикл. Коли знайдена підмножина, що містить єдиний цикл, витрати у якого менше або рівні нижнім межам для всієї решти підмножин, то такий цикл - оптимальний.

Підмножини циклів зручно представляти як вершини дерева, а процес розбиття як галуження (поява нових гілок) дерева. Тому метод названий методом «гілок і меж».

Згідно з цим методом множину всіх допустимих комбінацій розбивають на підмножини $Q_1, Q_2, \ldots, Q_{s_1}$. Кожну з останніх потім розбивають на підмножини $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_{s_1}}$, $\left(i = 1, 2, \ldots, s_1\right)$, і т. д., поки не дійдуть до окремих комбінацій. Цей процес галуження зручно зображати у вигляді дерева. На кожній підмножині $Q_{i_1, i_2, \ldots, i_s}$ визначають функцію переваги. Починаючи з вершини Q, пересуваються з етапу на етап, вибираючи кожного разу підмножину з мінімальним значенням цієї функції (або максимальним, якщо цього вимагає умова задачі). Одержане дерево буде деревом рішень.

Задача полягає у відповідному виборі функції переваги, щоб одержати найкращий розв'язок. Іноді вибір підмножини відбувається випадковим чином: при цьому вірогідність вибору даної підмножини тим більше, чим менше значення його функції переваги.

Основна проблема в цьому методі - вибір способу визначення нижньої (або верхньої) межі. Знайти досить точне її значення не завжди легко, та зате число даних гілок на дереві рішень в загальному випадку скорочується. Це випливає з того, що якщо значення нижньої межі для якої-небудь підмножини більше або рівне (у разі мінімізації; менше або рівно – в протилежному випадку) значенню функції одного з вже одержаних рішень, що мінімізується, то відповідну гілку дерева рішень виключають з розгляду.

2.3.2. Послідовні алгоритми оптимізації

Метод послідовного виключення можливостей – один з найпоширеніших способів прийняття рішень. Зводиться він до наступного виконання кроків [231]:

а) множина альтернатив розбивається на підмножини шляхом введення додаткових висловлень, що характеризують кожну окрему підмножину;

б) частину цих підмножин прагнуть виключити, використовуючи умови несумісності (логічної суперечності) висловлень, що відносяться до елементів підмножини, і висловлень, що характеризують вимоги до розв'язку;

в) з множинами, що залишилися, роблять аналогічну процедуру розбиття і виключення і т. д.

На основі досвіду математичного використання принципу послідовного виключення можливостей при розв'язанні ряду оптимізаційних задач планування і проектування в Інституті кібернетики ім. В. М. Глушкова НАН України в 60-х роках минулого століття була розроблена формалізована схема прийняття рішень, що одержала назву методу послідовного аналізу варіантів.

У основі методу послідовного аналізу варіантів лежить ідея представлення процесу розв'язання у вигляді багатоступінчатої структури, що нагадує структуру складного досвіду. Кожен ступінь пов'язаний з перевіркою наявності тих або інших властивостей у підмножини варіантів і веде або до безпосереднього скорочення початкової множини варіантів, або задає можливість такого скорочення в майбутньому. На основі теоретичного і практичного аналізу поставленої задачі спочатку потрібно чітко сформулювати, якими особливими властивостями повинен володіти шуканий варіант. Потім потрібно виявити по можливості більше ознак, які дозволяють встановити, що даний варіант не є шуканим. Серед цих ознак вибираються ті, що найлегше перевіряються і властиві одночасно, за можливістю, більшому числу варіантів. Після цього побудова числової схеми розв'язання полягає у виборі раціонального порядку перевірки ознак, що дозволяє в найкоротший час відсіяти непридатні варіанти і знайти оптимальний розв'язок.

Представлення процесу пошуку оптимального варіанту як послідовності складних дослідів нагадує формалізацію процесу ухвалення рішень на основі статистичних експериментів, розроблену А. Вальдом в теорії послідовних статистичних рішень.

У багатьох задачах для організації дослідів по звуженню множини можливих варіантів до шуканої множини вдається використовувати деякі загальні властивості оптимальних варіантів, що є узагальненнями принципу Р. Беллмана в динамічному програмуванні.

Загальна схема послідовного аналізу варіантів представляється таким чином.

Будується розв'язок багатоваріантної задачі як послідовний пошук достатньо звуженої підмножини варіантів на основі перевірки обмежень і обчислення критерію. Загальна схема такого пошуку може бути формалізована таким чином [231].

Нехай дано три множини: $W=\{\omega\}$ – множина варіантів, $\pi=\{\pi_\alpha\}$ – множина дослідів, $M=\{\alpha\}$ – множина індексів дослідів. У множині *M* виділена підмножина M^*, яку назвемо контрольною. Далі дана множина $I=\{\omega\}$, яку назвемо множиною результатів. Для кожного досліду π_α визначено в *I* підмножину $I_\alpha=\{\omega_\alpha^1,\omega_\alpha^2,\ldots\}$, кожен елемент якої назвемо результатом досліду π_α. У множині *I* виділена підмножина $\Omega=I$, на якій визначений оператор звуження $S(\omega)$, що ставить у відповідність кожному $\omega \in \Omega$ деяку підмножину $W_\omega = S(\omega)W$ з *W*. Ця відповідність природним чином розповсюджується на підмножини *U* з множини *W*:

$$S(\omega)U = U \bigcap W_\omega = U \setminus V_\omega,$$

де $V_\omega = W \setminus W'_\omega$.

На множині дослідів π визначений оператор реалізації *P*, що ставить у відповідність кожному $\pi_\alpha \in \Pi$ деякий елемент з $I_\alpha : P_{\pi_\alpha} = \omega^i_\alpha$, який назвемо реалізацією досвіду π_α.

Задача полягає у визначенні такої максимальної підмножини W^* з *W*, яка є інваріантною щодо будь-якої з контрольної множини M^* :

$$S\left(P_{\pi_\alpha}\right)W^* = W^* \text{ для кожного } \alpha \in M^*.$$

Вважатимемо, що множина результатів *I* містить елемент *e*, який має особливе значення. Розв’язком початкової задачі буде множина W_ω, що є звуженням вихідної множини *W*.

2.3.3. Методи побудови послідовності розв’язків

Метод побудови послідовності розв’язків застосовний до задачі

$$f\left(x^*\right) = \min_{x \in R \subset X} f\left(x\right), \tag{2.4}$$

якщо [231]:

а) можна знайти скінченне розширення $R^0 \supseteq R$ множини допустимих розв’язків *R* і функцію-міноранту $g(x)$, визначену на R^0 таку, що $g(x) \leq f(x)$ для всіх $x \subset R$;

б) можна побудувати алгоритм впорядкування φ, який на *k*-у кроці знаходить елемент $x_k \in R^0$, що має наступну властивість

$$g\left(x_k\right) = \min\left\{g\left(x\right) \mid x \in R^0 \setminus \left\{x_1, \ldots, x_{k-1}\right\}\right\}. \tag{2.5}$$

Інакше кажучи, повинен існувати ефективний метод φ побудови послідовності $\{x_i\}$ розв'язків з R^0 у порядку неспадання міноранти $g(x)$.

Аналогічно для задачі максимізації $f(x)$ на множині R повинен існувати метод φ побудови послідовності $\{x_i\}$ розв'язків з R^0 у порядку незростання мажоранти $g(x)$, тобто такої функції, що $g(x) \geq f(x)$ для всіх $x \in R$.

Критерій оптимальності. Якщо існує таке натуральне число k, що $R_k = \{x_1, \ldots, x_k\} \bigcap R \neq \varnothing$, $f\left(x_k^*\right) = \min\limits_{x \in R_k} f(x) \leq g(x_k)$, то x_k^* – оптимальний розв'язок задачі (2.4).

Методи розв'язання конкретних задач дискретної оптимізації, побудовані по вище зазначеній схемі, полягають в знаходженні за допомогою алгоритму φ послідовності x_i розв'язків множини R у порядку неспадання міноранти для виконання критерію оптимальності (2.4). Ефективність таких методів залежить від вибору розширення R^0 і міноранти $g(x)$. Якщо не існує число k, що задовольняє критерію (2.5), то метод перетворюється на повний перебір.

Вкажемо деякі варіанти схеми [231].

1. Якщо $g(x) = f(x)$ для всіх $x \in R$, тобто немає необхідності апроксимувати цільову функцію, критерій (2.4) спрацьовує, як тільки знайдено допустимий розв'язок $x^k \in R$.

2. Якщо $R^0 = R$, тобто множина допустимих розв'язків не розширюється, то, будуючи лише допустимі розв'язки $\{x_i\}$, завжди можна оцінити відхилення кращого з одержаних допустимих розв'язків x_k^* від оптимального x^*. Дійсно, якщо на k-у кроці умова (2.4) не виконується, то числа $g(x_k)$ і $f\left(x_k^*\right)$ є межами мінімального значення цільової функції $f(x)$ на R. При цьому з кожним кроком алгоритму межі уточнюються. Як наближений розв'язок можна вибрати x_k^* − кращий допустимий розв'язок з одержаних. Оскільки $f\left(x_k^*\right) - f\left(x^*\right) \leq f\left(x_k^*\right) - g(x_k)$,

то максимальне відхилення наближеного розв'язку x_k^* від оптимального x^* відомо.

Модифікуючи умову б) загальної схеми методів побудови послідовності розв'язків, одержимо наступні схеми [231]:

3. Нехай має місце припущення:

б') існує алгоритм, що визначає множину розв'язків

$$X = \left\{ x \mid f^- \le g\left(x\right) \le f^+, x \in R^0 \right\},$$

не обов'язково впорядкованих за збільшенням мінораnти. Тут f^-, f^+ – відповідно нижня і верхня межі для значень цільової функції на оптимальному розв'язку, наприклад $f^- = min\left\{ g\left(x\right) \mid x \in R^0 \right\}$. Оптимальний розв'язок визначається перебором розв'язків множини *X*. Ефективність даного варіанту залежить від числа елементів *X,* а точніше, від якості меж f^-, f^+.

4. Нехай має місце припущення:

б") існує алгоритм, який на *k*-у кроці виділяє множину розв'язків

$$H_k = \left\{ x \mid x \in R^0, f_k \le g\left(x\right) \le f_{k+1} \right\},$$

де f_k – зростаюча послідовність дійсних чисел.

Для послідовності $\left\{ H_i \right\}$ критерій оптимальності приймає наступний вигляд: якщо існує таке натуральне число *k,* що

$$R \bigcap \bigcup_{i=1}^{k} H_i \ne \varnothing$$

$$f\left(x_k^*\right) = \min_{x_k} \left\{ f\left(x_k\right) \mid x \in R \bigcap \bigcup_{i=1}^{k} H_k \right\} \le f_{k+1},$$

то x_k^* – оптимальний розв'язок задачі (2.4).

Особливість цього варіанту схеми полягає у тому, що на *k*-у кроці дописується до послідовності не один розв'язок з R^0, а ціла група їх.

2.3.4. Методи відтинання для розв'язування комбінаторних задач

Серед точних методів розв'язування задач дискретної оптимізації широкого розвитку і розповсюдження одержав метод відтинання, ідея якого вперше була запропонована Данцигом, а потім була розвинута в багатьох роботах, зокрема в роботах Гоморі.

Ефективність методу відтинання знаходиться в прямій залежності від ефективності способу побудови відтинань. Реалізація різних підходів до побудови відтинань породила велику кількість методів цієї групи.

Для групи методів відтинаючих площин використовується ідея «регуляризації» задачі. Вона полягає в зануренні початкової дискретної області допустимих розв'язків у відповідну неперервну опуклу область, тобто в тимчасовому відкиданні умов дискретності. Далі до одержаної регулярної задачі застосовуються стандартні методи оптимізації. Слід зазначити, що ефективність методу відтинання знаходиться в прямій залежності від ефективності способу побудови відтинань, а це викликає певні складнощі. Різні підходи до побудови відтинань, і різні модифікації методу відтинань для задач комбінаторної оптимізації розглянуто в роботах [91, 94, 98].

Розглянемо загальний опис простого алгоритму відтинаючих площин, що відноситься до класу методів, корисних для задач середнього розміру і цікавих з теоретичної точки зору. Розглянемо задачу цілочисельного програмування в стандартній формі:

$$\min c'x,$$

$$Ax = b, \tag{2.6}$$

$x \geq 0$ цілочислові змінні, де A, b, c – цілочислові.

Запишемо відповідну задачу лінійного програмування без обмежень на цілочисельність

$$\min c'x,$$

$$Ax = b, \tag{2.7}$$

$x \geq 0$ послабивши умову цілочисельності задачі (2.6).

Припустимо, що отримано розв'язок послабленої задачі, наприклад, за допомогою симплекс-алгоритму і знайшли базисний допустимий розв'язок x^*. Звичайно, у загальному випадку x^* не цілочисловий. Проте можливо одержати розв'язок початкової задачі, округлюючи координати отриманого розв'язку x^* до найближчих цілих чисел. Рис. 2.1 показує, чому такий підхід не спрацьовує: насправді біля x^* може взагалі не бути допустимих точок.

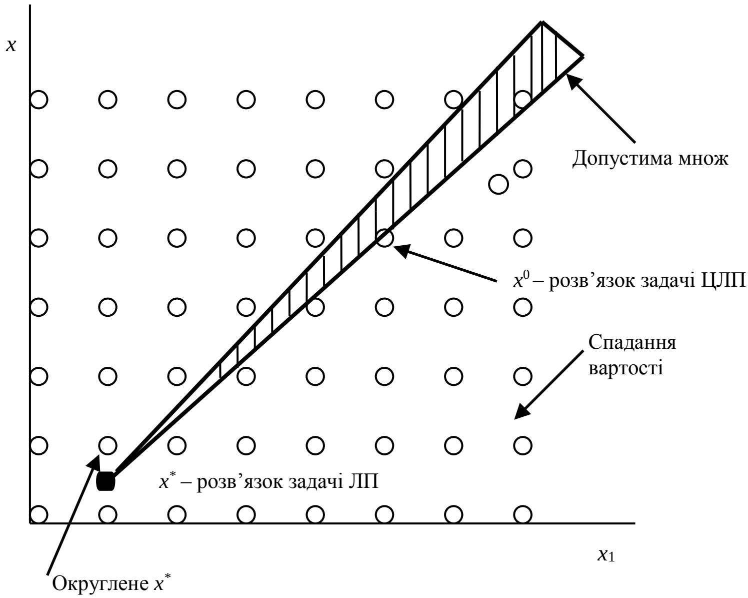


Рис. 2.1. Гіпотетична задача цілочислового лінійного програмування з оптимумом x^0 і її ослаблення з оптимумом x^*

Не важко бачити, що неперервний і дискретний оптимум розміщені досить далеко один від одного як за відстанню на площині, так і за оцінкою.

Слід зазначити два корисних факти. Перший: якщо неперервний оптимум x^* – розв'язок задачі лінійного програмування – виявляється цілим числом, то він розв'язує відповідну задачу цілочисельного лінійного програмування. Другий: оцінка $C\left(x^*\right)$

неперервного оптимуму є нижньою межею для $C\left(x^0\right)$ – оцінки дискретного оптимуму. Якщо до задачі цілочисельного лінійного програмування додати обмеження, що не виключає цілочислових допустимих точок, то розв'язок не змінюється. Отже, основна ідея алгоритму відтинаючих площин полягає в додаванні до задачі цілочисельного лінійного програмування таких лінійних обмежень по черзі, які б при розв'язанні відповідної задачі лінійного програмування, що є ослабленням початкової задачі, давали б цілочислові розв'язки.

Оскільки цілочислові допустимі точки не виключаються, то остаточний розв'язок ослабленої задачі цілочисельного лінійного програмування з доданими обмеженнями буде розв'язком початкової задачі. Описаний процес проілюстрований на рис. 2.2. На рис. 2.2(а) представлені початкова задача цілочислового лінійного програмування і неперервний оптимум x^*. На рис. 2.2(б) додане лінійне обмеження, що не виключає ніяких цілочислових допустимих точок, яке називається відтинаючою площиною (або просто відтинанням).

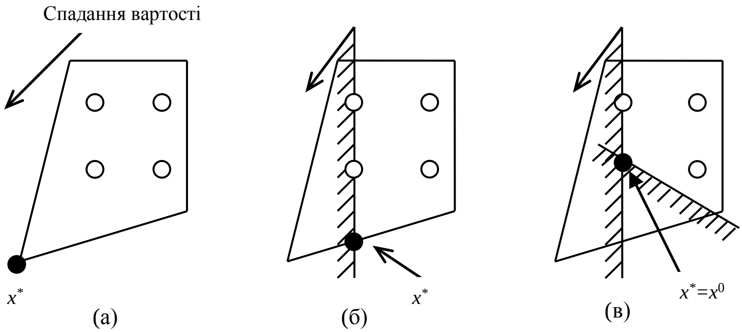


Рис. 2.2. Ілюстрація алгоритму відтинаючої площини:
а) неперервний оптимум x^*; б) новий розв'язок x^* після одного відтинання; в) розв'язок початкової задачі цілочисельного лінійного програмування після двох відтинань

При цьому відтинається частина допустимої множини, проте не втрачаються ніякі цілочислові точки. Новий неперервний оптимум зсувається, як показано на рисунку. На рис. 2.2(в) представлений результат додавання ще однієї відтинаючої площини; в результаті неперервний оптимум виявляється цілочисловим, і ця точка є розв'язком початкової задачі цілочисельного лінійного програмування.

М**етод відтинання** для комбінаторних задач з лінійними функціями цілі на комбінаторних множинах розроблений в Полтаві під керівництвом доктора фізиком-математичних наук Ємця О. О. та апробований його учнями на різних комбінаторних множинах [89, 90]. Для комбінаторних задач суть даного методу полягає в наступному: 1) здійснюється релаксація задачі комбінаторної оптимізації: умова дискретності (належності розв'язку комбінаторній множині) замінюється умовою неперервності – належності опуклій оболонці комбінаторної множини, розглядається знову ж таки неперервна задача; 2) розв'язується симплекс-методом задача лінійного програмування на новій заданій області; 3) для отриманої точки перевіряється виконання всіх обмежень системи комбінаторного многогранника. Якщо точка задовольняє умову належності комбінаторній множині E, то задача розв'язана. Інакше здійснюється побудова відтинання, якому знайдена точка не задовольняє, у вигляді

$$\sum_{j=1}^{s} \alpha_{m+1,j} z_j \leq b_{i+1},\ i \in J_p,$$

і знову розв'язується задача лінійного програмування, поки не буде знайдена точка, що є вершиною комбінаторного многогранника, а отже і елементом комбінаторної множини.

Треба зауважити, що область визначення комбінаторних задач – скінченна множина, що визначена обмеженнями комбінаторного многогранника, а тому замкнена, і тому через скінченну кількість кроків метод відтинання закінчує свою роботу, тобто знайдеться точка, яка є розв'язком вихідної задачі.

Слід зазначити, що ефективних і одночасно універсальних способів для розв'язання екстремальних оптимізаційних задач не існує. Але, навіть якщо б вони і існували, то не завжди витрати на пошук оптимального розв'язку виправдовують вигоди від його застосування, і часто в практичних ситуаціях віддають перевагу швидкому вибору розв'язку, що близький до оптимального. Тому разом з точними методами, що забезпечують справжній екстремум, розвиваються різноманітні наближені методи, в яких пошук розв'язку тим або іншим способом обмежується.

2.3.5. Евристичні алгоритми

Особливий клас алгоритмів складають так звані евристичні алгоритми. Сюди входять наближені алгоритми, направлені на розв'язання однієї якої-небудь задачі.

Це пов'язано з тим, що точні методи оптимізації вимагають, як правило, дуже великого об'єму обчислень. Тому має велике практичне значення створення простих методів, що дають розв'язки, достатньо близькі до оптимальних. Методи, в яких не оцінюється близькість одержаних розв'язків до оптимальних, прийнято називати евристичними. Якщо ж можна оцінити відхилення розв'язків від оптимальних, то такий метод називають наближеним. Евристичний метод після його успішного теоретичного дослідження може перейти в розряд наближених.

Найчастіше в евристичних методах застосовують локальну оптимізацію.

Нехай маємо множину комбінацій $P = \{\pi_1, \pi_2, \ldots, \pi_N\};$ ставиться задача обчислення мінімуму функції, визначеної на цій множині, і відшукання комбінацій, на яких цей мінімум досягається. Метод локальної оптимізації полягає в наступному.

Для кожної комбінації $\pi_i \in P$ визначимо множину Q_i комбінацій, які називатимемо сусідніми з π_i. Сусідство розуміється в широкому значенні: так для множини перестановок сусідніми можуть вважатися дві з них, коди яких відрізняються однією транспозицією; для підмножини вершин цілочислової багатовимірної решітки сусідніми можуть вважатися дві вершини, що відрізняються на одиницю тільки в одній координаті і т. д. В термінах теорії графів це можна трактувати так: кожній комбінації $\pi_i \in P$ поставимо у відповідність деяку вершину *i* графа *G*. Сусідні вершини *i* та *j* графа *G* з'єднаємо ребром $(i, j)(\pi_j \in Q_i)$. Одержаний граф G_1 з множиною ребер *U* називатимемо графом сусідства комбінацій, або графом 1-сусідства, тобто сусідства на відстані одного ребра. У графі G_1 може існувати сукупність *Z* його вершин, в якій

$$F\left(\pi_i\right) = F\left(\pi_j\right), \text{ якщо } \pi_i, \pi_j \in Z$$

$$F\left(\pi_i\right) < F\left(\pi_j\right), \text{ якщо } \pi_i \in Z; \pi_j \notin Z; \left(i, j\right) \in U.$$

Таку множину Z назвемо ізольованою. Ця множина, зрозуміло, може складатися і з однієї вершини. Перший етап локального підходу, про який іде мова, полягає в тому, щоб, вибравши довільну комбінацію, визначити для неї граф сусідства, після чого на цьому графі для всіх $\pi_j \in Q_i$ визначити значення $F\left(\pi_j\right)$.

Другий етап полягає в операції, яку прийнято називати спуском. Знаходимо $\pi_{j_o} \in Q_i$ так, щоб

$$F\left(\pi_{j_0}\right) = \min_{\pi_j \in Q_i} F\left(\pi_j\right),$$

і якщо $F\left(\pi_{j_0}\right) < F\left(\pi_i\right)$, то переходимо до комбінації π_{j_o}.

Так за скінченну кількість кроків дійдемо ізольованої множини. Проте навіть попадання на ізольовану множину не дає упевненості, що досягнуто не локальний екстремум (в даному випадку, мінімум), а глобальний. Тому локальні «проби» доводиться продовжувати. Одним із способів продовження пошуків може бути вибір нової допустимої комбінації і повторення операції спуску в графі, який є об'єднанням нового графу сусідства і старого. Можливі і інші способи. Визначимо, наприклад, послідовність графів сусідства $G_1, G_2, \ldots, G_s$ (графи, у яких сусідство на відстані не більше 1, 2,…, s ребер). Після попадання в G_1 на ізольовану множину Z вершин переходимо до графа G_2 з довільної вершини $\pi_i \in Z$. Якщо Z і в G_2 залишається ізольованим, перейдемо до G_3 і т. д. Якщо ж з вершини π_i, можливий спуск в ізольовану множину Z_1, то, перейшовши в яку-небудь вершину $\pi_j \in Z_1$, повернемося до графа G_1 і повторимо процес вже з іншої вершини G_1.

У описаному підході є багато невизначеного. Особливо неясно, як будувати графи сусідства і чим ця побудова буде кращою за випадковий вибір комбінацій. Проте досвід підказує, що одержувати сусідні комбінації іноді легше, ніж будувати випадкові послідовності. Аналіз конкретної задачі часто приводить до способу визначення сусідніх комбінацій, що забезпечує ефективний спуск до оптимального або достатньо кращого розв’язку.

Якщо допустима область R розв’язання задачі містить декілька точок, що реалізовують екстремум (мінімум або максимум) функції $f(x)$, то можна розглядати дві задачі оптимізації: визначення локального і глобального екстремумів. Перша з цих задач простіша, і часто її рішення практично прийнятне. Тому в даний час методи локальної оптимізації набули широкого поширення, особливо в застосуванні до складних оптимізаційних задач, тобто до задач великої розмірності, цільові функції і функції додаткових обмежень яких не мають властивостей, що дозволяють побудувати ефективні алгоритми визначення глобального екстремуму.

Локальний алгоритм може бути охарактеризований двома параметрами: величиною околу, що вивчається на кожному кроці, і числом ознак, що запам’ятовується про кожен елемент.

2.3.6. Метод вектора спаду

Метод вектора спаду є сімейством алгоритмів, об’єднаних загальною схемою. Ці алгоритми призначені для розв’язання дискретних оптимізаційних задач і дають локальний розв’язок [153, 155, 231].

Метод вектора спаду − ітераційний метод направленого перебору, що належить до типу градієнтних методів, розроблений і багатократно апробований в Інституті кібернетики ім. В. М. Глушкова НАН України під керівництвом академіка Сергієнка І. В. його учнями [226, 231].

Нехай X − деякий комбінаторний метричний простір з метрикою r. На підмножині $R \subseteq X$ визначена функція $f(x)$. Тоді загальна задача в таких позначеннях може бути сформульована таким чином. Визначити елемент $x \in R \subset X$, що доставляє екстремум функції.

У застосуванні до цієї задачі схема методу вектора спаду полягає в наступному.

1. Визначення початкового значення x_0; $x_i = x_0$.

2. Побудова множини $L_\rho(x_i)\bigcap R$, де $L_\rho(x_i)$ − окіл радіусу ρ з центром в точці x_i.

3. Розв'язок (точний або наближений) локальної задачі

$$f(x_{i+1}) = \underset{x\in L_\rho(x_i)\bigcap R}{ext} f(x).$$

4. Порівняння x_i і x_{i+1}. Якщо $x_i = x_{i+1}$, то обчислення припиняємо, приймаючи за локальний розв'язок задачі x_i; інакше переходимо на п. 2 алгоритму, замінюючи i на $i+1$ і т. д. до тих пір, поки на деякому кроці i^* не знайдеться $x_{i^*} = x_{i^*+1}$.

Ця схема є деякою сукупністю алгоритмів.

Дійсно: 1) $\rho = \rho(i)$; зокрема, можна прийняти $\rho = const$, допустима також багатозначність $\rho(i)$, що дозволяє уточнювати одержані розв'язки; 2) локальна задача може розв'язуватися різними методами.

Зазначимо, що ця схема в застосуванні до кожної конкретної задачі (класу задач) повинна враховувати її конкретні властивості (класу задач), тому метод в застосуванні до вужчого класу задач вимагає спеціальних додаткових досліджень, що враховують властивості простору R, його метрики і множини, а також властивості функції $f(x)$.

2.4. Екстремальні задачі оптимізації при умові багатокритеріальності та методи їх розв'язання

Результати дослідження практичних задач планування та керування показують, що в реальній постановці ці задачі є задачами з декількома цільовими функціями. Вираз «досягти максимального ефекту при найменших витратах» означає ухвалення рішення при двох критеріях. Оцінка діяльності підприємств і планування як системи прийняття рішень визначається на основі більше десятка критеріїв: виконання плану виробництва за об'ємом, за номенклатурою, плану реалізації, прибутку за показниками рентабельності, продуктивності праці і т. д. Отже, практично будь-яка прикладна задача є багатокритеріальною, яку звести до одного критерію, як правило, досить складно, оскільки цілей може бути значно більше. В цьому випадку оптимізація проводиться за декількома частковими критеріями, і така задача в загальному випадку і називається задачею багатокритеріальної оптимізації.

Отже, багатокритеріальна задача це задача про оптимізацію двох і більше критеріїв, яка має реальний економічний зміст і відіграє важливу роль в економіці, плануванні виробництва та ін. Одним з важливих застосувань такої задачі може бути задача про досягнення мінімуму собівартості деякої продукції, визначення максимуму рентабельності виробництва, розрахунку прибутковості і взагалі про оптимізацію деяких відносних показників якості (трудомісткості, продуктивності і т. п.). Такі моделі відображають тенденцію постійного зниження рівня собівартості в розрахунку на одиницю продукції та підвищення якісних показників виробництва при збільшенні масштабів виробництва.

В зв’язку з цим особливе значення на даний час набуває теорія прийняття рішень при наявності багатьох критеріїв. Одним із основних понять теорії багатокритеріальної оптимізації є поняття оптимального за Парето, або ефективного розв’язку. Вперше проблема оптимізації векторного критерію була сформульована економістом Парето в 1896 р.

Проблема прийняття рішень в економіці, зокрема оптимального планування, виникає в силу двох принципових обставин: з одного боку, багатоваріантності планових рішень, з іншого − цілеспрямованості економічних систем. Множина альтернативних варіантів плану визначається наявними можливостями економічного розвитку; вибір із цієї множини визначається цілями планованої системи. Прийняте рішення є результатом спільного розгляду цілей і можливостей, їх узгодження одна з одною.

При використанні математичних методів в аналізі та прийнятті планових рішень обидві складові задачі вибору повинні знайти адекватне відображення в економіко-математичній моделі. Не зупиняючись тут на принципах опису множини допустимих варіантів плану, звернемося до методів математичного моделювання цілей економічного розвитку.

У найбільш простій і досить поширеній інтерпретації ціль розуміється як деякий заздалегідь визначений стан, досягнення якого задане в деякій розглянутій системі. Таке трактування, зрозуміло, досить обмежене, оскільки за його межами повністю залишається задача вибору цілі, що складає зміст цільової стадії планування.

Отже, нехай елементи системи цілей представлені частковими критеріями $f_1,\dots, f_l : X \to R;$ тут і далі X − множина допустимих планів, що ототожнюється зі своїм економіко-математичним описом і знаходиться у просторі R^n. Функції $f_1,\dots, f_l$ формують векторний критерій оптимальності $F(x)=\left(f_1(x),\dots, f_l(x)\right);$ вважається, що j-й цілі відповідає максимізація складової $f_j(x)$. Пара (F,X), утворена векторним критерієм $F : X \to R^l$ і множиною X, представляє собою модель, або задачу, багатокритеріальної (векторної) оптимізації.

Таким чином, векторний критерій оптимальності виступає як засіб для розв'язання існуючого протиріччя між складністю цілей економічних систем і обмеженими можливостями їх безпосереднього економіко-математичного моделювання. Разом з тим було б помилковим повністю пов'язувати виникнення багатокритеріальних задач зі специфікою математичного методу дослідження економічних проблем. Множинність цілей економічних систем має об'єктивний характер і знаходить своє модельне відображення у формі векторного критерію.

Часткові цілі, представлені компонентами векторного критерію, поєднані різного роду зв'язками. Зокрема, цілі пов'язані тим, що висувають вимоги до одних і тих же варіантів допустимих планів. Такі зв'язки опосередковані обмеженнями, що формують допустиму множину X, і умовно можуть бути названі внутрішніми.

У свою чергу, зовнішні зв'язки відображають порівняльну важливість цілей, їх нагальність, взаємозамінимість та ін. Ці зв'язки обумовлені структурою системи цілей, об'єктивно існуючими відношеннями між її елементами. Звичайно, зовнішні зв'язки опосередковані досить складними соціально-економічними взаємодіями й тому важко піддаються економіко-математичному опису. Незважаючи на те, що такі зв'язки є найважливішими реальностями задачі прийняття планових рішень, вони ніяк не відображені векторним критерієм і тому виходить, залишаються за межами багатокритеріальної моделі.

Тоді природно, що така модель, яка розглядається сама по собі, виявляється недостатньою для обґрунтованого вибору оптимального варіанта плану. Часткові критерії $f_1, \ldots, f_l$, будучи в загальному випадку різнонаправленими, оптимізуються на множині *X* різними розв'язками рівнянь типу

$$x^j \in arg\ max\left\{ f_j\left(x\right) | \, x \in X \right\},\ j \in N_l.$$

Кожний із цих розв'язків може виявитися неоптимальним, а найчастіше просто незадовільним з погляду інших компонентів векторного критерію *F*. Отже, виникає характерна для багатокритеріальних задач проблема досягнення компромісу між частковими цілями або, говорячи інакше, узгодження цих цілей. Таке узгодження вимагає порівняння різних елементів системи цілей, їх зіставлення однієї з іншою, що неможливо без належного урахування всієї сукупності як внутрішніх, так і зовнішніх зв'язків між ними.

Отже, можна констатувати, що векторний критерій $F = \left(f_1, \ldots, f_l \right)$, навіть якщо його компоненти поставлені у відповідність всім складовим системи цілей, не є її еквівалентом і повинен бути посилений додатковою інформацією. До надходження такої інформації багатокритеріальна модель $Z\left(F, X\right)$ дозволяє робити висновок про порівняльну перевагу різних варіантів плану та про можливості вибору тих або інших альтернатив у тій мірі, у якій часткові критерії не суперечать один одному.

В багатокритеріальній задачі оптимізації порівняння розв'язків здійснюється за допомогою заданих на множині допустимих альтернатив числових функції $f_1, \ldots, f_s, f_{s+1}, \ldots, f_m$, що називаються критеріями. Передбачається, що $m \geq 2$, бо при $m = 1$ задача оптимізації являється однокритеріальною.

В більшості випадках при дослідженні проблеми багатокритеріальності всі критерії, крім одного, обраного домінуючого, приймалися як обмеження, оптимізація проводилася за домінуючим критерієм. Такий підхід до розв'язання практичних задач значно знижує ефективність прийнятих рішень. У зв'язку із цим у даній роботі розглядається загальна постановка задачі багатокритеріальної (векторної) оптимізації і деякі основні підходи до її розв'язання при умові комбінаторних властивостей розв'язків.

В останні роки методам розв'язання багатокритеріальних задач присвячена велика кількість літератури та інтерес до цієї області продовжує зростати. Важливою особливістю сучасного етапу розвитку даного наукового напрямку є більш широке використання методів багатокритеріальної оптимізації для розв'язання практичних задач.

Методи багатокритеріальної оптимізації являють собою чисельну реалізацію певного правила вибору ефективної (Парето-оптимальної), слабо-ефективної (оптимальної за Слейтером), строго-ефективної (оптимальної за Смейлом) альтернатив. Вони класифікуються за типами інформації наступним чином [24]: методи багатокритеріальної оптимізації; методи, які не використовують інформацію про перевагу на множині критеріїв; методи, які використовують один тип інформації про перевагу на множині критеріїв; методи, які використовують різні типи інформації про перевагу на множині критеріїв; спеціальні методи.

Одним з найбільш розповсюджених методів розв'язання багатокритеріальних задач є метод зведення багатокритеріальної задачі до однокритеріальної шляхом згортання векторного критерію в суперкритерій. При цьому кожний критерій множиться на відповідний йому ваговий коефіцієнт (коефіцієнт важливості). Також серед відомих методів слід назвати такі як принцип справедливого компромісу, принцип слабкої оптимальності за Слейтером, принцип наближення за всіма локальними критеріями до ідеального розв'язку, метод квазіоптимізації локальних критеріїв (метод послідовних поступок).

Отже, в даному розділі розглянуто методи розв'язування екстремальних комбінаторних однокритеріальних та багатокритеріальних задач. Слід зазначити, що задачі багатокритеріальної оптимізації були розглянуті в багатьох роботах, але без врахування комбінаторних властивостей області допустимих значень, що не завжди адекватно описують моделі прикладних задач. Хоча, можна навести ряд прикладів, де множини – області допустимих розв'язків таких задач – мають переставні, сполучні та ін. властивості області допустимих значень, тобто ці задачі потрібно розглядати як екстремальні задачі на комбінаторних конфігураціях з багатьма критеріями. В наступних розділах розглядаються підходи до розв'язування задач комбінаторної оптимізації з багатьма критеріями на основі існуючих методів, а також пропонуються нові підходи до розв'язання екстремальних комбінаторних задач.

Висновки до розділу

У даному розділі розглядаються екстремальні задачі дискретної оптимізації та методи їх розв'язання: робиться загальний огляд стану проблеми та напрямів досліджень. Проведено аналіз актуальності теми та важливість обраного напряму досліджень. Формулюється загальна постановка задачі та здійснюється огляд існуючих методів її розв'язання. Саме формулювання екстремальних комбінаторних задач диктує вибір операцій, залучених для їх розв'язання. Тому необхідно вміти робити генерування множини елементів комбінаторної конфігурації і мати у своєму розпорядженні відповідну множину значень функції $f(x)$. Важливим є методика порівняння значень функції і виділення з них максимального або мінімального. Операція переліку практично рідко виявляється здійсненною, тому що число можливих комбінацій може бути занадто великим. При розв'язанні комбінаторних задач часто використовуються такі добре відомі конструкції з елементів скінченної множини, як сполучення, розміщення, перестановки й т. п. В розділі визначаються поняття цих конструкцій через формалізацію комбінаторних схем з використанням поняття відображення однієї скінченної множини X в іншу Y. На основі цього підходу побудовані комбінаторні конфігурації, які відповідають найпростішим комбінаторним об'єктам: розміщенням, перестановкам, комбінаціям, розбиттям та іншим об'єктам, а також визначені їх властивості для подальшого використання при розв'язуванні задач.

РОЗДІЛ 3. МЕТОДИ ГЕНЕРУВАННЯ КОМБІНАТОРНИХ КОНФІГУРАЦІЙ ТА ПОБУДОВА ЇХ ГРАФІВ

В даному розділі пропонується новий метод розв'язування екстремальних задач на комбінаторних конфігураціях. Оскільки задачі розглядаються на різних комбінаторних конфігураціях, які відповідають найпростішим об'єктам: розміщенням, перестановкам, розбиттям, сполученням, то при їх розв'язуванні досить важливою є методика перебору елементів конфігурацій, тобто їх генерування та порівняння, а також дослідження всіх елементів деякого класу комбінаторних об'єктів. Комбінаторні алгоритми призначаються для виконання обчислень на дискретних скінченних математичних структурах, що представляють собою комбінаторні конфігурації. Слід зазначити, що існує велика кількість методів генерування комбінаторних об'єктів, в [148] розглядається три методи генерування таких комбінаторних об'єктів, як перестановки, розміщення, сполучення – лексикографічний, антилексикографічний, метод генерування за мінімальне число транспозицій. В усіх цих методах елементарною операцією, яка застосовується до елементів комбінаторної конфігурації є поелементна транспозиція, тобто обмін значеннями змінних. В [11] крім вище зазначених методів генерування розглядаються метод транспозиції суміжних елементів та метод заміни одного елемента іншим.

В розділі представлені нові методи генерування, що будують послідовності елементів конфігурацій з допомогою однієї транспозиції двох елементів (сусідніх чи не сусідніх), а також підхід, що грунтується на переміщенні максимального елемента множини, з якої утворюється конфігурація. В залежності від тих чи інших методів генерування елементів комбінаторних конфігурацій будуються підходи до розв'язування екстремальних задач оптимізації. Зокрема, представлено новий метод направленого структурування, що поєднує засоби комбінаторного аналізу і теорії графів. Основна ідея цього методу перекликається з відомим методом послідовного аналізу варіантів, що описаний в першому розділі, розглядаються загальні підходи побудови алгоритмів та ефективні методи розв'язування екстремальних задач на різних конфігураціях, серед яких найбільша увага приділяється методам роботи з великими масивами даних, алгоритмам генерування і сортування комбінаторних об'єктів, а також алгоритмам на графах, що отримали широкий розвиток та застосування.

Графи досить тісно пов’язані з поняттям комбінаторних конфігурацій: перестановок, розміщень, сполучень, розбиттів та ін. Випуклими оболонками таких комбінаторних конфігурацій є комбінаторні многогранники, властивості яких відіграють важливу роль для розв’язування задач оптимізації, тому розглянемо деякі поняття з теорії графів.

3.1. Використання теорії графів для розв’язування екстремальних комбінаторних задач

У термінах теорії графів формулюється велика кількість задач, пов’язаних з дискретними об’єктами [11, 62, 86, 87]. Такі задачі виникають під час проектування інтегральних схем, схем керування, у дослідженні автоматів, в економіці й статистиці, теорії розкладів і дискретній оптимізації. Очевидно, із всіх математичних об’єктів графи займають одне з перших місць як формальні моделі реальних систем. Графи знайшли застосування практично у всіх галузях наукових знань: фізиці, біології, хімії, математиці, історії, лінгвістиці, соціальних науках, техніці й т. п. Найбільшою популярністю теоретико-графові моделі користуються при дослідженні комунікаційних мереж, систем інформатики, хімічних і генетичних структур, електричних ланцюгів й інших систем мережної структури.

До типових задач теорії графів й їх додатків можна віднести наступні: задача про найкоротший ланцюг, заміна обладнання, складання розкладу руху транспортних засобів, розміщення пунктів швидкої допомоги, розміщення телефонних станцій, задача про максимальний потік, задача про розміщення та покриття, розфарбування в графах, зв’язність графів і мереж (проектування найкоротшої комунікаційної мережі, синтез структурно-надійної мережі циркуляційного зв’язку, аналіз надійності стохастичних мереж зв’язку), структурний синтез лінійних виробничих ланцюгів, покриття схеми заданим набором типових підсхем та ін.

Для екстремальних комбінаторних задач важливість застосування теорії графів пояснюється тим, що комбінаторні конфігурації, на яких представляються задачі, можна представити у вигляді графів.

Графи комбінаторних конфігурацій мають багато цікавих властивостей; при їх вивченні виникають задачі, що представляють інтерес не тільки для теорії графів, комбінаторики, топології і геометрії, але і для теорії лінійного програмування.

Використання властивостей графів комбінаторних конфігурацій можуть послужити підвищенню ефективності «традиційних» і розробці нових методів розв'язування екстремальних задач комбінаторної оптимізації. Комбінаторні моделі можуть бути застосовані для представлення оптимізаційних задач, що виникають при оптимальному розміщенні на графах. Теорія графів вивчає екстремальні властивості графів, розглядаючи множину його граней всіх розмірностей як деякий комплекс. Але при розв'язанні таких задач виникають проблеми, пов'язані з складністю математичних моделей, великим об'ємом інформації і т. д., оскільки більшість задач на комбінаторних конфігураціях є *NP*-повними. Більшість задач на графах стосується визначення компонент зв'язності, пошуку маршрутів, відстаней і т. п. Проте при розв'язанні прикладних задач відповідні їм графи, що залучаються, мають достатньо великий об'єм, а аналіз можливий лише із застосуванням сучасної обчислювальної техніки. В даному розділі досліджується екстремальна задача на комбінаторних конфігураціях. На підставі встановленого взаємозв'язку між задачами на комбінаторних множинах і графами многогранників комбінаторних множин вивчаються деякі структурні властивості допустимої області, а також сформульовано ряд тверджень, які дають можливість побудувати методи розв'язання комбінаторної задачі з використанням графів. Зокрема, в роботі [60] описаний спосіб побудови гамільтонового шляху усередині гіперграней. У статті [61] ставиться задача побудови гамільтонового шляху між гіпергранями, тобто, як побудувати гамільтонів шлях, додавши до двох графів без спільних вершин третій граф, що не має спільних вершин з першими двома, якщо гамільтонів шлях побудовано для перших двох графів. Для подальшого викладу матеріалу сформулюємо деякі необхідні визначення і властивості.

Як було зазначено в попередньому розділі, досить важливим є комбінаторне представлення визначених класів дискретних оптимізаційних задач в різних інтерпретаціях, що дає можливість більш адекватно описати постановку прикладної моделі.

Отже, розглядаємо граф – як фігуру, яка складається з непорожньої скінченної множини V точок (вершин) і скінченної множини E орієнтованих чи неорієнтованих ліній (ребер), що з'єднують деякі пари вершин.

Нехай v_1, v_2 вершини графа, $e = (v_1, v_2)$ – ребро, що їх сполучає. Тоді вершина v_1 і ребро *e* інцидентні, ребро *e* і вершини v_1, v_2 також інцидентні. Два ребра, інцидентні одній вершині, є суміжними; дві вершини, інцидентні одному ребру, називаються також суміжними.

Надалі, якщо не вказано інше, вершини графа *G* позначатимемо буквою v з індексами чи без: v, v_2, v_{34}, множину вершин графа – *V;* ребра – буквою *e* з індексами чи без: e, e_6, e_{97}, відповідно множину ребер графа $G - E$. Ребро, що з'єднує деяку вершину саму з собою, називають петлею. Ребра, що з'єднують одну й ту саму пару вершин, називають мультиребрами. Граф, що не містить мультиребер та петель, називають звичайним графом. Граф, в якому допускаються мультиребра чи петлі, називають мультиграфом. Орієнтоване ребро $e = (v_1, v_2)$ називається дугою, яка виходить з вершини v_1 і заходить в вершину v_2.

Зазначимо, що в подальшому розглядаються графи, які не є мультиграфами, тобто такі графи що не містять петель.

Визначення 3.1. Граф, усі ребра якого неорієнтовані, називають неорієнтованим графом, а граф, усі ребра якого орієнтовані – орієнтованим графом, або орграфом.

Підграфом графа $G = \langle V, E \rangle$ називається граф G' для якого виконуються наступні умови $V' \subseteq V$ і $E' \subseteq E,$ де V', V – множина вершин E', E – множина ребер.

Маршрутом у довільному графі, що починається у вершині v_1 і закінчується у вершині v_2, називають послідовність вершин та ребер вигляду:

$$v_1 e_{i_1} v_{i_1} e_{i_2} v_{i_2} e_{i_3} \ldots v_{i_{n-1}} e_{i_n} v_2,$$

де всі сусідні елементи інцидентні.

Маршрут в неорієнтованому графі називається ланцюгом.

Шляхом в орієнтованому графі називається односторонній орієнтований маршрут.

Очевидно, що маршрут і для звичайних графів однозначно визначається послідовністю вершин: $v_1 v_{i_1} v_{i_2} \ldots v_{i_{n-1}} v_2$.

Маршрут, який не містить повторень вершин і ребер, крім, можливо, двох крайніх вершин v_1 та v_2, називають простим.

Замкнений шлях (ланцюг) $(v_1 = v_2)$ називають контуром (циклом).

Для подальшого викладу підходу до розв'язання задачі розглянемо поняття про гамільтонові та напівгамільтонові шляхи.

Визначення 3.2. Гамільтоновим шляхом у графі називають простий шлях, який проходить всі вершини графа.

Розглянемо деякі прикладні задачі, що формулюються в термінах теорії графів.

Знаменита задача про комівояжера є комбінаторною задачею на множині перестановок і пов'язана із знаходженням гамільтонового циклу найкоротшої сумарної довжини. Як відомо, в цій задачі вважається, що є n міст, які неодмінно повинен відвідати комівояжер, – всі і рівно по одному разу, пройшовши загальний шлях найменшої сумарної протяжності. Якщо між містами є дороги, то вони інтерпретуються як ребра графа порядку n з вказаною довжиною. Вершини такого графа – вершини переставного многогранника, і є містами.

Задача комівояжера є приклад комбінаторної задачі на графах і має велике практичне і теоретичне значення. Через свою обчислювальну складність вона рівносильна цілому класу переборних задач і часто використовується математиками для порівняльного аналізу і вивчення складності алгоритмів оптимізації в дискретній математиці.

Деякі прикладні задачі, що формулюються в термінах теорії графів представлено в [140, 177].

Задача 3.1. Нехай задано N міст, між кожною парою міст необхідно забезпечити автомобільне сполучення. Ця мета досягається завдяки побудові мережі доріг між деякими безпосередньо близькими містами. Вартість будівництва доріг відома, необхідно для досягнення мети витратити мінімум ресурсів. Задача зводиться до пошуку мінімального остовного дерева, що визначається вартістю будівництва. Та ж задача може бути інтерпретована, як прокладання комп'ютерної мережі, при мінімальній витраті кабелів.

Задача 3.2. Припустимо, необхідно найняти людей, які виконують певну роботу. Кожна людина може працювати в різний час, і брати за цю роботу різну кількість грошей. Необхідно найняти працівників так, щоб від моменту А до моменту Б хоча б один з працівників виконував цю роботу, і витратити на це мінімум грошей. Це задача на визначення найкоротшого шляху. Для її розв'язання необхідно побудувати мультиграф. Вершини – це моменти від А до Б тобто час, в який працівник може заступити на роботу і піти з неї, а ребра – це різні працівники Необхідно відшукати мінімальний шлях від А до Б. Причому, граф орієнтований.

Задача 3.3. (про зайнятість працівників) Нехай є n працівників і m видів робіт. Кожен працівник може виконувати якусь роботу, і кожну роботу можуть виконувати різні люди. Необхідно розподілити робітників так, щоб максимальна кількість була зайнята, причому одну роботу може виконувати один робітник. Будуємо граф, де певного робітника сполучаємо з деяким видом робіт, які він може виконувати, також створюємо псевдоджерело, і сполучаємо його із усіма працівниками. Всі роботи сполучаємо з псевдостоком. Вага дуг, що сполучають працівника і роботу рівна 1. Тоді максимальна кількість зайнятих людей рівна пропускній спроможності графа, який носить назву мережі.

Також можуть бути сформульовані задачі складання розкладу, аналізу мереж в електротехніці, аналізу ланцюгів Маркова в теорії ймовірностей, в програмуванні, в проектуванні електронних схем, в економіці, в соціології і т. ін. Досить важливим при розв'язуванні вище сформульованих задач є властивості множини вершин графа, а також співвідношення між графами та підграфами.

Надалі поняття теорії графів використаємо для представлення графів комбінаторних конфігурацій, граф комбінаторної конфігурації – це головний об'єкт, який надалі буде використовуватися для побудови нового методу направленого структурування.

3.2. Методи генерування комбінаторних конфігурацій

Класичною задачею комбінаторики є задача визначення числа способів розміщення деяких об'єктів в якійсь кількості «ящиків» так, щоб були виконані задані обмеження. Цю задачу можна сформулювати більш формально таким чином [148]: дані множини X, Y, причому $|X| = n,\ |Y| = m,$ визначити скільки існує функцій $f : X \to Y,$ що задовольняють заданим обмеженням?

Елементи множини X відповідають об'єктам, елементи множини Y – ящикам, а кожна функція $f : X \to Y$ визначає деяке розміщення, указуючи для кожного об'єкту $x \in X$ ящик $f(x) \in Y$, в якому даний об'єкт знаходиться.

Без втрати загальності можемо завжди вважати, що $X = 1,\ldots,n$ і $Y = 1,\ldots,m$. Кожну функцію f можна тоді ототожнити з послідовністю $\langle f(1),\ldots,f(n)\rangle$.

Якщо $|X| = n$, $|Y| = m$, то число всіх функцій $f : X \to Y$ дорівнює m^n і таким чином визначається множина розміщень.

Легко знайти кількість розміщень, для яких кожен ящик вміщує не більше одного об'єкту – такі розміщення відповідають взаємно однозначним функціям. Позначимо $[n]_m$ число всіх взаємно однозначних функцій з n-елементної множини в m-елементну множину. Легко показати, що

$$[n]_m = n(n-1)\ldots(n-m+1). \qquad (3.1)$$

Якщо $m = n$, то кожна взаємнооднозначна функція $f : X \to Y$ є взаємно однозначним відображенням множини X на множину Y. У такому разі $[n]_n = n(n-1)\cdot\ldots\cdot 1$ позначаємо $n!$. Кожне взаємнооднозначне відображення $f : X \to X$ називається перестановкою множини X.

В більшості комбінаторних методів для їх подальшої реалізації і застосування до розв'язування задач є важливим встановлення початкових умов тобто, множини комбінаторних об'єктів, що задовольняють деяким властивостям. Для цього є потреба в генеруванні і створенні таких комбінаторних об'єктів.

В попередньому розділі було розглянуто властивості комбінаторних множин як об'єктів – підмножин заданої дискретної множини. На основі даних властивостей можна формувати і розглядати елементи комбінаторних множин як вершини графів та деяких многогранників. Далі розглянемо відомі методи генерування комбінаторних множин [11, 148] та опишемо нові, для подальшого їх використання в комбінаторних методах. Для початку, як приклад, розглянемо перестановки, що найчастіше використовуються при моделюванні прикладних задач.

3.2.1. Методи генерування перестановок

Розглянемо перестановку як впорядковану множину з n елементів, які різняться між собою порядком їх розміщення. Число перестановок множини $\{a_1, a_2, \ldots, a_n\}$ різних елементів дорівнює $n!$. Відомо, що такі перестановки ізоморфні перестановкам індексів цих елементів. Тому будемо вважати, що елементами множини перестановок є числа від 1 до n.

Таке спрощення буде можливо також провести і для інших комбінаторних об'єктів, про які буде далі йти мова.

Розглянемо для подальшого використання та розв'язування задач три відомих методи генерування послідовності всіх $n!$ перестановок n-елементної множини, що описані в [11, 148, 184].

Лексикографічний метод формування перестановок. Послідовність перестановок на множині $\{1, 2, \ldots, n\}$ представлена в лексикографічному порядку, якщо вона записана в порядку зростання чисел. В загальному випадку якщо $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ і $\tau = (\tau_1, \tau_2, \ldots, \tau_n)$ − перестановки, то вважаємо, що σ лексикографічно менше τ, тоді і тільки тоді, якщо для деякого $k > 1$ маємо $\sigma_j = \tau_j$ для всіх $j < k$ і $\sigma_k < \tau_k$. Легко точно визначити відповідність між цілими числами $0, 1, 2, \ldots, n! - 1, n!$ та перестановками із множини $\{1, 2, \ldots, n\}$, записаними в лексикографічному порядку. Позначимо перестановку $p = (\pi_1, \pi_2, \ldots, \pi_n)$.

Перестановки можна породжувати одну за іншою таким чином: починаючи з початкової перестановки $p = (\pi_1, \pi_2, \ldots, \pi_n)$ від неї переходимо до іншої шляхом перегляду p справа наліво у пошуку найправішої позиції, в якій $\pi_i < \pi_{i+1}$. Знайшовши її, тепер шукаємо π_j, найменший елемент, розташований праворуч від π_i і більший за нього; потім здійснюється транспозиція елементів π_j і π_i і відрізок $\pi_{i+1}, \ldots, \pi_n$ (елементи якого розташовані в порядку спадання) перевертається. Таким чином одержуємо нову перестановку, з якою також виконуємо вище зазначені дії.

Загальну ефективність алгоритму за лексикографічним впорядкуванням, що означає запис слів в алфавіті, визначають числом порівнянь між елементами перестановки.

Зазначимо, що в лексикографічно впорядкованій послідовності перестановок існує повна підпослідовність $k!$ перестановок із k правих елементів, в якій решта елементів не переміщується. Це спостереження дозволяє отримати рекурентне співвідношення для числа транспозицій $(\pi_r \leftrightarrow \pi_s)$ або $(\pi_i \leftrightarrow \pi_j)$ і числа порівнянь $(\pi_i > \pi_{i+1})$ або $(\pi_1 > \pi_j)$ відповідно використаних для породження перших $k!$ із $n!$ перестановок.

Зокрема, зазначимо, що для кожного із n можливих значень π_1 тобто для кожної підпослідовності перестановок, відповідних $n-1$ найправішим компонентам, використовується I_{n-1} транспозицій. Трансформація останньої перестановки однієї з цих підпослідовностей в першу перестановку наступної підпослідовності вимагає $\left[(n+1)/2\right]$ перестановок, і всього таких перетворень буде $n-1$. Таким чином, за винятком перетворення, яке здійснюється при $i=0$, існує $nI_{n-1}+(n-1)\left[(n+1)/2\right]$ транспозицій, а тому

$$I_n = nI_{n-1} + (n-1)\left[\frac{n+1}{2}\right] \text{ і } I_1 = 0 .$$

Слід зазначити, що лексикографічний порядок не може породжуватися так ефективно, як інші порядки, перетворення однієї перестановки в іншу в цьому випадку – це складний процес. Важливість лексикографічного порядку визначається його простотою і природністю.

Розглянемо ефективність цього лексикографічного породження перестановок у вигляді алгоритму, для цього представимо перестановки в вигляді масивів. Припустимо, що елементи множини перестановок запам'ятовуються у вигляді елементів масиву $P=\{P[1],\ldots,P[n]\}$. Елементарною операцією, яка застосовується до масиву P є поелементна транспозиція, тобто обмін значеннями змінних $P[i]$ і $P[j]$ де $1 \le i, j \le n$. Ця операція згідно методів математичного програмування позначається через $P[i] :=: P[j]$. Очевидно, що вона еквівалентна послідовності команд

$$d := P[i]; P[i] := P[j]; P[j] := d,$$

де d − деяка допоміжна змінна.

Тоді метод лексикографічного впорядкування легко зрозуміти, якщо елементами, які переставляються, взяти числа $1, 2, \ldots, n$. На множині всіх послідовностей довжини n з елементами з множини $X = \{1, \ldots, n\}$ визначається лексикографічний порядок:

$$\langle x_1, x_2, \ldots, x_n \rangle < \langle y_1, y_2, \ldots, y_n \rangle \Leftrightarrow \exists k \geq 1 \left(x_k \leq y_k \wedge (x_i = y_i) \forall i < k \right).$$

Зазначимо, що якщо замість чисел $1, 2, \ldots, n$ взяти букви $a, b, \ldots, z$, то лексикографічний порядок визначає стандартну послідовність, в якій слова довжини n з'являються в словнику.

Антилексикографічний порядок визначається подібним чином до лексикографічного порядку $<'$, з тією різницею, що як черговість позицій в послідовності, так і впорядкування елементів множини X зворотні по відношенню до початкових даних, що визначені згідно лексикографічного порядку:

$$\langle x_1, x_2, \ldots, x_n \rangle <' \langle y_1, y_2, \ldots, y_n \rangle \Leftrightarrow \exists k \geq 1 \left(x_k > y_k \wedge (x_i = y_i) \forall i < k \right).$$

Як приклад розглянемо перестановки множини $X = \{1, 2, 3\}$ в лексикографічному (а) і антилексикографічному (б) порядку:

(a)	(b)
1 2 3	1 2 3
1 3 2	2 1 3
2 1 3	1 3 2
2 3 1	3 1 2
3 1 2	2 3 1
3 2 1	3 2 1

Рис. 3.1. Перестановка множини

Алгоритм генерування перестановок в антилексикографічному порядку сформулювати набагато зручніше: 1) спочатку кожну перестановку лексикографічного порядку записати в зворотному вигляді; 2) всі перестановки записати в оберненому порядку від $n!$ до 1. Відмітимо, що послідовність перестановок множини $\{1,\ldots,n\}$ в цьому випадку має наступні властивості:

1) в першій перестановці елементи йдуть в тій послідовності, що зростає, а в останній – в спадаючій; іншими словами, остання перестановка є протилежна першій;

2) у останній позиції, визначають послідовність перестановок множини $\{1,\ldots,n\}\setminus\{p\}$ в антилексикографічному порядку.

Зазначимо, що число транспозицій при генеруванні кожної наступної перестановки є величина змінна: кожна друга перестановка вийшла за рахунок однієї транспозиції $P[i]:=:P[j]$, але разом з ними є і такі, які вимагають $\lfloor(n-1)/2\rfloor+1=\lfloor(n+1)/2\rfloor$ транспозицій. Хоча середнє число транспозицій, що припадає на кожну перестановку, невелике, проте в деяких випадках кращим був би алгоритм, в якому кожна наступна перестановка утворюється з попередньої за допомогою виконання однієї транспозиції. Це досить суттєво, коли з кожною перестановкою пов’язані деякі обчислення і коли існує можливість використання часткових результатів, одержаних для попередньої перестановки, якщо послідовні перестановки мало відрізняються одна від одної.

Основну схему алгоритму можна описати за допомогою наступної процедури, що полягає в генеруванні всіх елементів перестановок $P[1],\ldots,P[n]$ через послідовну генерацію всіх елементів перестановок $P[1],\ldots,P[n-1]$ і заміну елементу $P[n]$ на один з елементів $P[1],\ldots,P[n-1]$.

Розглянемо метод генерування всіх перестанонок за мінімальне число транспозицій. Даний метод генерування перестановок будує послідовність, в якій різниця між двома послідовними перестановками ще менше ніж в двох попередніх: кожна наступна утворюється з попередньої за допомогою одноразової транспозиції сусідніх елементів [148].

(a)				$(б)$				$(в)$			
1	2	3	4	1	2	3	4	1	2	3	4
2	1	3	4	2	1	3	4	2	1	3	4
1	3	2	4	2	3	1	4	2	3	1	4
3	1	2	4	3	2	1	4	2	3	4	1
2	3	1	4	3	1	2	4	3	2	4	1
3	2	1	4	1	3	2	4	3	2	1	4
1	2	4	3	4	3	2	1	3	1	2	4
2	1	4	3	3	4	2	1	1	3	2	4
1	4	2	3	3	2	4	1	1	3	4	2
4	1	2	3	2	3	4	1	3	1	2	4
2	4	1	3	2	4	3	1	3	4	1	2
4	2	1	3	4	2	3	1	3	4	2	1
1	3	4	2	4	1	3	2	4	3	2	1
3	1	4	2	1	4	3	2	4	3	1	2
1	4	3	2	1	3	4	2	4	1	3	2
4	1	3	2	3	1	4	2	1	4	3	2
3	4	1	2	3	4	1	2	1	4	2	3
4	3	1	2	4	3	1	2	4	1	2	3
2	3	4	1	4	2	1	3	4	2	1	3
3	2	4	1	2	4	1	3	4	2	3	1
2	4	3	1	2	1	4	3	2	4	3	1
4	2	3	1	1	2	4	3	2	4	1	3
3	4	2	1	1	4	2	3	2	1	4	3
4	3	2	1	4	1	2	3	1	2	4	3

Рис. 3.2. Послідовності перестановок, що одержані різними способами

На рис. 3.2 послідовності перестановок, одержані за допомогою методів: а) лексикографічний; б) антилексикогра-фічний; в) метод генерування за мінімальне число транспозицій.

Даний метод приписується Джонсону і Троттеру [148]. Суть його розглянемо на прикладі. Припустимо, що вже побудована послідовність перестановок елементів $2, 3, \ldots, n,$ що має властивість згідно методу, наприклад: 23, 32 для $n = 3$. Тоді необхідну послідовність перестановок елементів $1, 2, \ldots, n$ одержимо, вставляючи елемент 1 всіма можливими способами в кожну перестановку елементів $2, 3, \ldots, n$. Одержимо:

$$\{1\,2\,3\}, \{2\,1\,3\}, \{2\,3\,1\}, \{3\,2\,1\}, \{3\,1\,2\}, \{1\,3\,2\}.$$

У загальному вигляді елемент 1 розташовується між першою і останньою позиціями по черзі вперед і назад $(n-1)!$ раз. На основі цієї конструкції можна легко одержати рекурсивний алгоритм, що генерує необхідну послідовність перестановок для довільного n. Але слід зазначити, що даний метод має недолік: послідовність перестановок будується разом вся і лише після закінчення побудови всієї послідовності елементів перестановки, її можна зчитувати, а тому розв'язування таким способом потребує великого об'єму пам'яті. Розглянемо нерекурсивний варіант цього алгоритму. У цьому варіанті для кожного $i, 1 \le i \le n,$ булева змінна $PR[i]$ містить інформацію про те, чи переноситься елемент i вперед ($PR[i]$ – істина) або ж назад ($PR[i]$ – хибна), змінна $C[i]$ показує, яку з можливих $n-i-1$ позицій елемент 1 займає щодо елементів $i+1, \ldots, n$ на своєму шляху вперед або назад. Позицію елементу 1 визначаємо на підставі його позиції в блоці, що містить $i, i+1, \ldots, n$, а також на підставі числа елементів з $1, 2, \ldots, i-1$, які знаходяться зліва від цього блоку. Це число, буде значенням змінної x, обчислюється як число елементів $j < i$, які б рухаючись назад, досягли б крайнього лівого положення. Кожна нова перестановка утворюється транспозицією найменшого із елементів j, який не знаходиться в граничному положенні з його лівим або правим сусідом.

Інтерпретацію послідовностей, одержаних за допомогою вище описаних методів генерування перестановок розглянемо за допомогою графа G_n. Вершини графа G_n можна розглядати як перестановки множини $\{1,\dots,n\}$, в якому дві вершини, відповідних перестановок f і g, сполучені ребром, тоді і тільки тоді, коли g утворюється з f одноразовою транспозицією сусідніх елементів. Якщо графу G_n надати орієнтацію, визначивши орієнтовані ребра та з множини вершин – набір упорядкованих пар різних вершин, то в графі G_n можна визначити шлях – маршрут, в якого всі вершини різні.

Для перестановок можна побудувати замкнутий маршрут, який проходить через кожну перестановку один раз і визначає гамільтонів шлях.

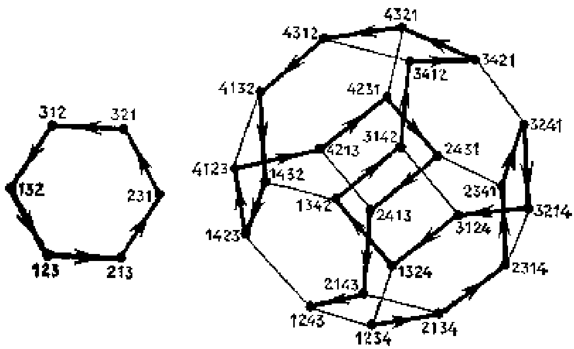


Рис. 3.3. Інтерпретація послідовності перестановок, одержаної за допомогою методу генерування переміщення за мінімальне число транспозицій

На рис. 3.3 стрілки вказують напрямок і послідовність обходу по вершинах графа – перестановках, що одержана за допомогою методу генерування всіх перестанонок за мінімальне число транспозицій. Дана послідовність відповідає гамільтоновому шляху в графі.

Вище описано огляд відомих методів генерування множини перестановок, кожний з яких має як свої переваги так і свої недоліки. Надалі визначимо нові методи генерування комбінаторних конфігурацій, які краще враховують специфіку розглядуваних задач і позбавлені тих недоліків, що мають відомі методи, а також дають можливість побудувати і представити граф комбінаторної конфігурації, який можна орієнтувати за значеннями цільвої функції.

Рекурсивний метод побудови перестановок – полягає в тому, що один елемент фіксується в кінці перестановки, а останні перебираються. Тобто, нехай маємо послідовність чисел $P=\left(p_1, p_2, \ldots, p_n\right)$, для якої виконуюється умова $p_1 \leq p_2 \leq \ldots \leq p_n$. Тоді на першому етапі фіксується останній елемент – p_n, всі інші $\left(n-1\right)$-елементів $p_1, p_2, \ldots, p_{n-1}$ перебираються рефлексивно. Наступний крок полягає в тому, що елемент p_{n-1} поміщається на останню позицію і фіксується, а набір $\left(n-1\right)$ елементів послідовності *P*, що складається з $p_1, p_2, \ldots, p_{n-2}, p_n$ переставляється. Для подальшого розуміння знову замість змінних елементів $p_1, p_2, \ldots, p_n$ візьмемо числа $1, 2, \ldots, n,$ множину яких позначимо N_n. Тоді побудова послідовності елементів перестановок здійснюється за формулою:

$$P_n\left(N_n\right)=\bigcup_{i=n}^{1} P_{n-1}\left(N_n \setminus i\right) i . \qquad (3.2)$$

Отже, даний метод полягає в виконанні послідовності кроків:

1. Визначаємо послідовність чисел і впорядковуємо її за зростанням $p_1 \leq p_2 \leq \ldots \leq p_n$. Присвоюємо $i:=n$.

2. Фіксуємо елемент з індексом *i* в визначеній послідов-ності (на першому етапі це буде елемент – p_n). останні $\left(n-1\right)$-елементів генеруємо наступним чином:

а) шляхом переходу зліва направо знаходимо найлівішу позицію в якій $p_i > p_{i+1}$;

б) знайшовши її, шукаємо p_j, найменший елемент, розташований ліворуч і менший за нього;

в) здійснюємо транспозицію елементів p_j і p_i − отримаємо новий набір елементів, що визначає перестановку.

3. Зменшуємо на одиницю $i:=i-1$, переходимо на крок 2.

Отже, шляхом послідовного генерування всіх елементів $p_{i_1}, p_{i_2}, \ldots, p_{i_{n-1}}$ при фіксованому останньому елементу p_{i_n} отримаємо всі перестановки. Таким чином, можемо виписати всі перестановки *n* елементів систематичним перебором всіх можливих для $p_n, p_{n-1}, \ldots, p_1$. Простий спосіб зробити це полягає в тому, щоб почати з перестановки $(1, 2, \ldots, n)$ і послідовно зміщувати на одну позицію всі $(n-1)$ елементів. Зміщення по циклу на один розряд елементів приводить до наступної породженої перестановки.

Зазначимо, що послідовності перестановок, одержані за допомогою вище зазначеного методу, можна інтерпретувати як граф G_n, вершини якого відповідають всім елементам множини перестановок $P(A)$.

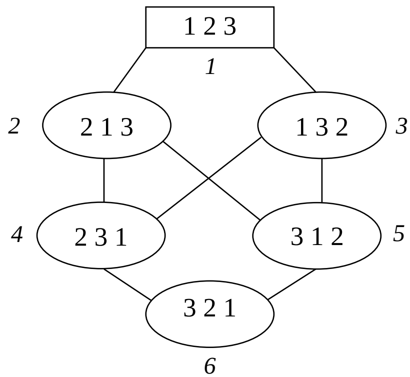


Рис. 3.4. Інтерпретація послідовності перестановок для $n = 3$

У графі дві вершини, що відповідають перестановкам p_1 і p_2, сполучені ребром тоді і тільки тоді, коли p_2 утворюється з p_1 одноразовою транспозицією двох елементів. при фіксованому останньому.

Розглянемо тепер детально структуру відповідних графів для невеликих значень n. Для $n = 3$ на рис. 3.4 зображено неорієнтований граф, що відображає утворення перестановки p_i з перестановки p_j рекурсивним методом.

Якщо кількість елементів множини, з якої утворюється перестановка, збільшується, то один з них фіксуємо, а останні переставляємо таким же методом. Для наглядної ілюстрації розглянемо метод генерування на прикладі $n = 4$, представивши всі елементи перестановки як розкладання графа множини перестановок на підграфи, де виділемо деякий фіксований елемент. Перший підграф представляє ті елементи конфігурації перестановок, в яких на останньому четвертому місці стоїть максимальний елемент – цифра 4. Підграф при $x_4 = 4$ представлений на рис. 3.5а якщо об’єднати перестановки, в яких на четвертому місці стоїть цифра 3, то одержимо підграф, представлений на рис. 3.5б. Аналогічно можна представити графи елементів конфігурації перестановок, де на останньому місці зафіксовані цифри 2 і 1 відповідно.

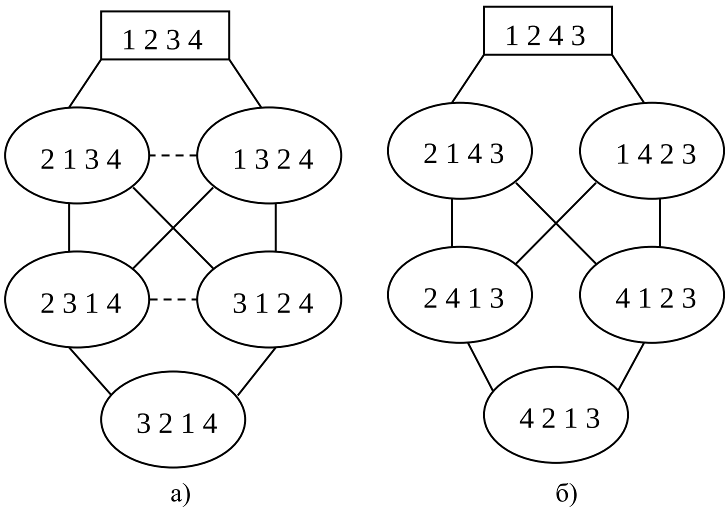


Рис.3.5. Розкладання на підграфи послідовності перестановок для $n = 4$

Для представлення загального графа позначимо підграф, що представлений на рис. 3.5а – *A*. Таким чином, можна зобразити всі підграфи, які одержані з початкового *A* за допомогою однієї транспозиції. Слід відзначити, що у всіх підграфах однозначно визначається місце останнього елементу. І взагалі, всі підграфи є копією(проекцією) підграфа *A*, так що їх можна в тому ж порядку розташувати один під одним. Неважко помітити, що підграф *B* одержується з підграфа *A*, якщо в останньому у всіх перестановках зробити транспозицію чисел 4 і 3. Аналогічно підграф *C* отримаємо як проекцію підграфа *B*, якщо в останньому зробити транспозицію (3, 2), а підграф *D* з підграфа *C* після транспозиції чисел 2, 1. враховуючи загальні поняття теорії графів та об’єднавши всі підграфи, одержимо загальний граф перестановок для $n = 4$: $G_n(P) = A \bigcup B \bigcup C \bigcup D$ зображений на рис. 3.6.

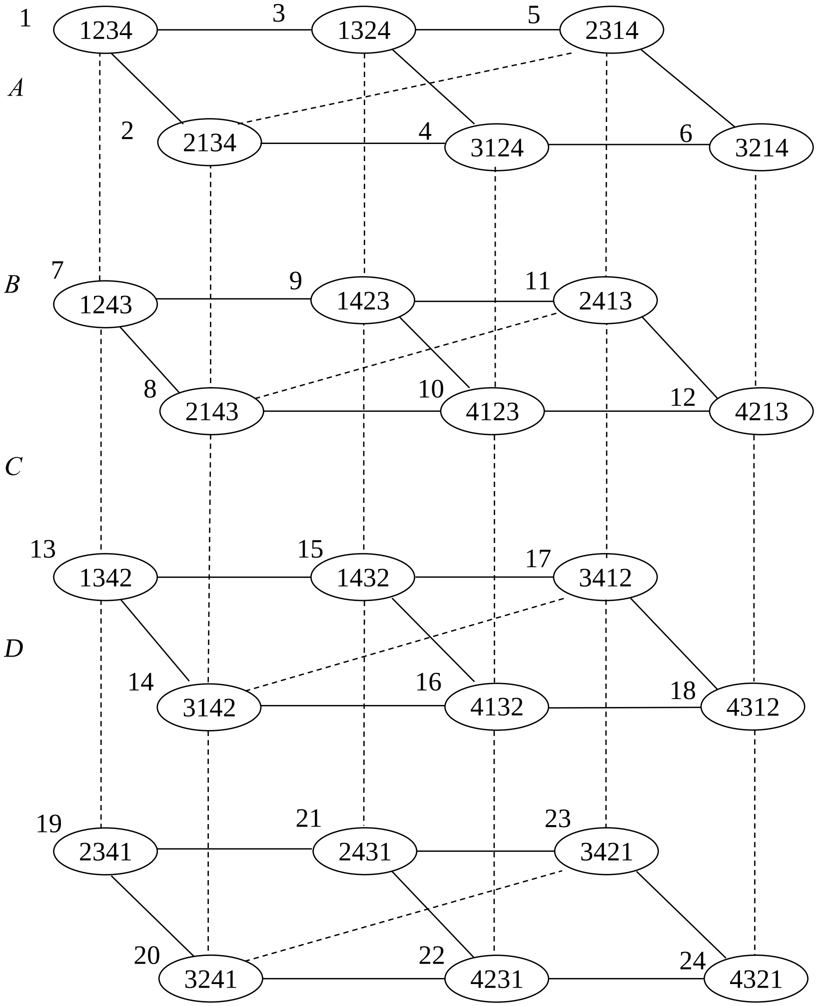


Рис. 3.6. Граф перестановок $G_n(P)$

Оскільки при цьому в підграфі $A(B, C)$ робиться одна транспозиція, то всі вершини копії з'єднуються ребром з відповідними вершинами оригіналу. В результаті одержуємо так званий ***граф перестановок,*** вершинами якого є елементи перестановки, і дві вершини в ньому суміжні, якщо відповідні перестановки відрізняються однією транспозицією елементів.

Побудову графа перестановок можна здійснити по індукції наступним чином. Спочатку для $n=2$ дві перестановки (12) та (21) як дві вершини з'єднаємо ребром. Переходимо до $n=3$. Для цього допишемо в кожній з двох вершин з правого боку число 3. На рис. 3.4 отримаємо вершини *1* і *2*. Тепер зробимо копію цього графа (цих вершин); в ньому виконаємо транспозицію ($3 \Leftrightarrow 2$); З'єднуємо ребром відповідні вершини оригіналу та копії. На рис. 3.4 отримаємо вершини *3* і *5*. Нарешті робимо копію вершин *3* і *5,* виконуємо на них транспозицію ($2 \Leftrightarrow 1$) і з'єднуємо ребром відповідні вершини першої та другої копій. Тим самим отримали граф перестановок $G_3(P)$. Далі зрозуміло: якщо побудовано граф перестановок $G_{n-1}(P)$, то дописуємо в кінці коду перестановок цього графа число *n*, отримуємо підграф A_1. Якщо отримано підграф $A_i(1 \le i < n)$, то робимо його копію, виконуємо на вершинах копії транспозицію ($n-i+1 \Leftrightarrow n-i$), з'єднуємо ребром відповідні вершини оригіналу та копії. Тим самим отримуємо підграф A_{i+1}. Продовжуємо вказані дії до отримання підграфу A_n. Це і завершує побудову графа перестановок $G_n(P)$. На рис. 3.6 підграфи $A_i\,(i=1, 2, 3, 4)$ відповіно позначені *A, B, C, D*.

Подібне представлення генерування комбінаторної конфігурації дає можливість розглядати екстремальні комбінаторні задачі на графі деякої множини. Таким чином надалі можемо оперувати з відомою структурою графа конфігурацій, а також надати йому орієнтацію в залежності від вигляду функції, яка оптимізується в екстремальних задачах.

Метод переміщення максимального елементу

Даний метод генерування елементів комбінаторної конфігурації передбачає вибір максимального елементу в множині

$P=(p_1,p_2,\dots,p_n)$ з якої генеруються елементи комбінаторного об'єкту і його переміщення справа наліво. Переваги даного методу не є абсолютними по відношенню до інших графів перестановок, але в залежності від типу функції такий граф може сприяти побудові досконалішого алгоритму розв'язку поставленої задачі

Звичайно перестановку можна представити за допомогою таблиці з двома рядками, з яких перший містить всі елементи множини X в заданому порядку (як правило, в зростаючому), а другий – елементи $f(x)$, які розташовуються під відповідним елементом x.

Так як природа елементів множини X для подальших досліджень неістотна – приймемо для простоти $X=\{1,\dots,n\}$, так само, як і в попередніх випадках, позначимо множину всіх перестановок цієї множини через P_n, а довільна перестановка $f\in P_n$ ототожнюватиметься з послідовністю $\langle a_1,\dots,a_n\rangle$, де $a_i=f(i)$.

Визначимо суперпозицією перестановок f і g, під якою розумітимемо перестановку fg, що визначається таким чином:

$$fg(i)=f(g(i)).$$

Зазначимо, що для суперпозиції двох перестановок

$$f=\begin{pmatrix}1&2&3&4&5\\5&3&2&1&4\end{pmatrix}\ g=\begin{pmatrix}1&2&3&4&5\\2&5&3&1&4\end{pmatrix},$$

достатньо змінити порядок стовпців в перестановці f так, щоб в першому рядку одержати послідовність, що є в другому рядку перестановки g, тоді другий рядок перестановки f дає суперпозицію fg.

Зазначимо, що кожна перестановка $f\in P_n$ однозначно визначає перестановку f^{-1}, таку що $ff^{-1}=f^{-1}f=e$, де перестановку

$$e=\begin{pmatrix}1 & 2 & \dots & n\\ 1 & 2 & \dots & n\end{pmatrix}$$

називатимемо тотожною перестановкою. Очевидно, що $ef = fe = f$ для довільної перестановки $f \in S_n$. Називатимемо f^{-1} перестановкою, зворотною до *f*. Щоб її визначити, достатньо поміняти місцями рядки в записі перестановки *f*. Тоді для довільних перестановок $f, g, h \in S_n$ виконуються умови $(fg)h = f(gh)$.

Щоб відобразити цей факт, говоритимемо, що S_n утворює групу щодо операції суперпозиції. Цю групу називатимемо симетричною групою степеня *n*.

Довільну перестановку, що є циклом довжини 2, називатимемо транспозицією. Важливу роль в подальших міркуваннях виконуватимуть транспозиції сусідніх елементів, тобто транспозиції вигляду $[i, i+1]$.

Метод переміщення максимального елемента також зводиться до побудови підграфів меншого розміру, які відрізняються один від одного положенням елемента *n*. Тоді послідовність таких перестановок будується за допомогою наступної формули:

$$P_n(N_n) = \bigcup_{i=n}^{1} P_{n-1}(N_n \setminus n;\, n \to i\text{ - му місті}). \tag{3.3}$$

Для більш наглядної ілюстрації розглянемо приклад для кількості елементів множини n=5, з якої формуються об'єкти перестановок. Нехай задано множину елементів перестановки $P = (p_1, p_2, \dots, p_n)$, які визначені числами $(1, 2, 3, 4, 5)$. На початковому етапі роботи методу вибирається початковий елемент, що є максимальним – в даному випадку 5, фіксується його позиція на останньому місці справа. Розглянемо граф формування підпослідовності перестановок при фіксованому останньому елементі $x_5 = 5$.

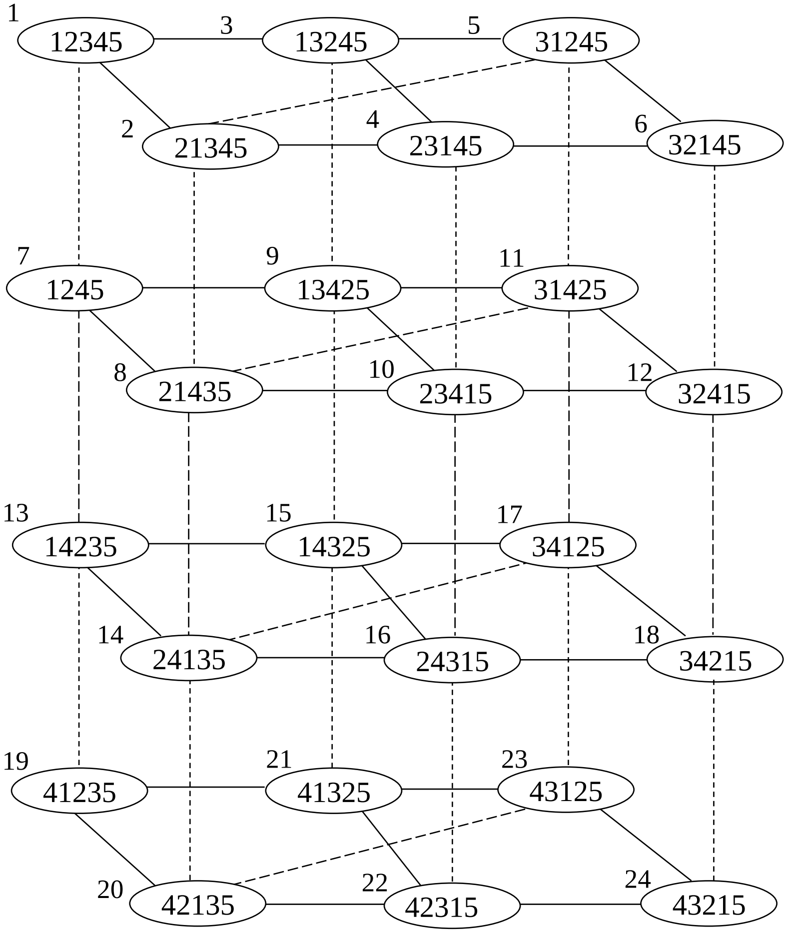


Рис. 3.7. Підпослідовність перестановок при фіксованому значенні $x_5 = 5$

Далі максимальний елемент переміщується на одну позицію вліво, тобто займає четверте місце. Тоді наступну підпослідовність перестановок можна представити у вигляді графа на рис. 3.8.

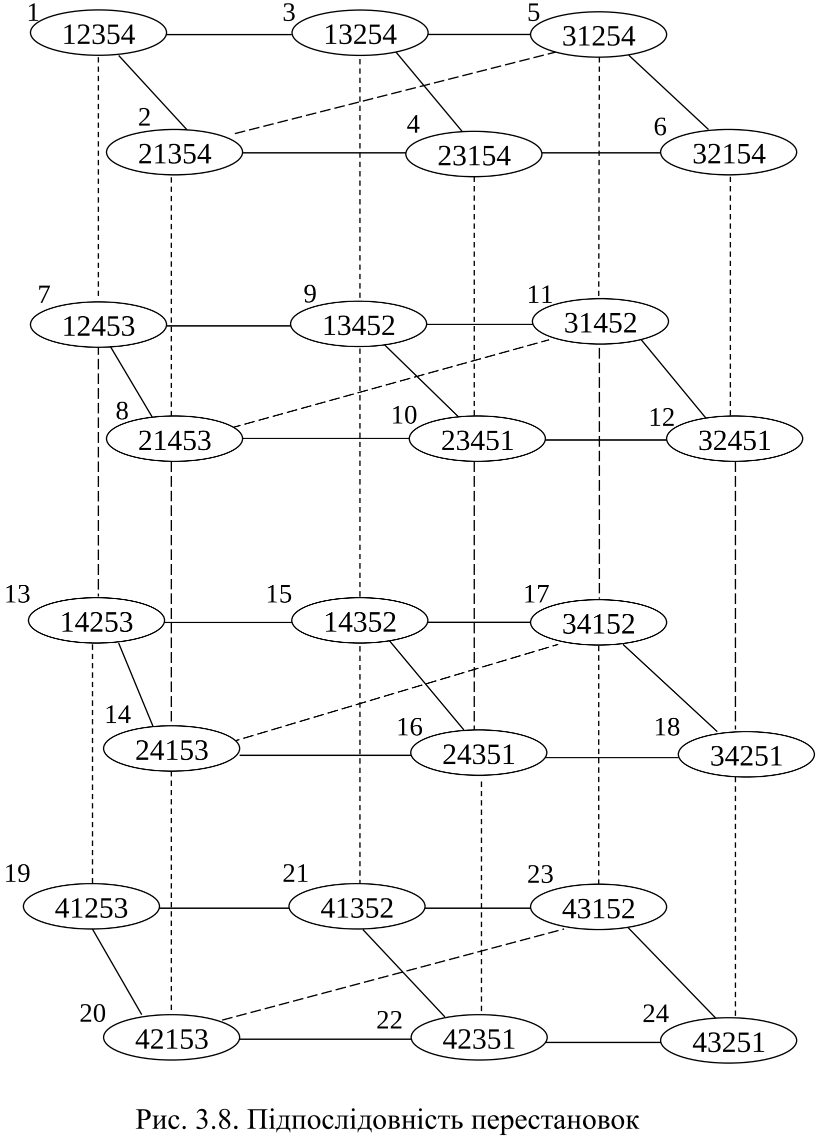


Рис. 3.8. Підпослідовність перестановок
при фіксованому значенні $x_4 = 5$

Наступна позиція, яку займає максимальний елемент визначається $x_3 = 5$. Тоді підпослідовність зображується у вигляді наступного графа.

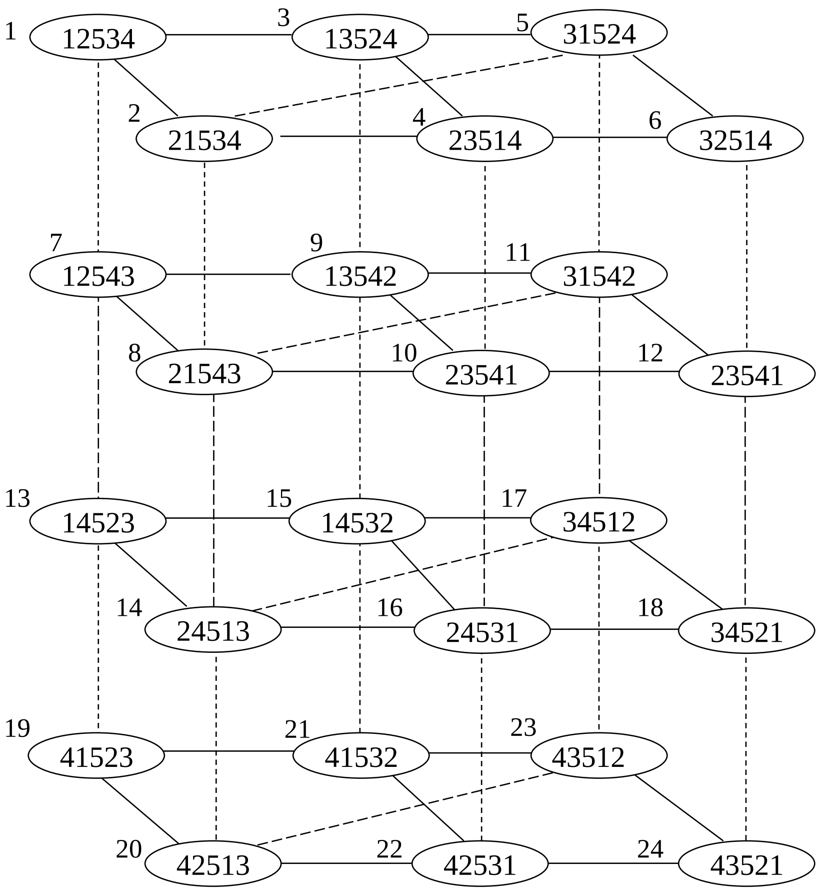


Рис. 3.9. Підпослідовність перестановок при фіксованому значенні $x_3 = 5$

Відповідно далі зображаються графи, на яких представлені підпослідовності перестановок, що утворюються при позиції максимального елементу $x_2 = 5$, $x_1 = 5$.

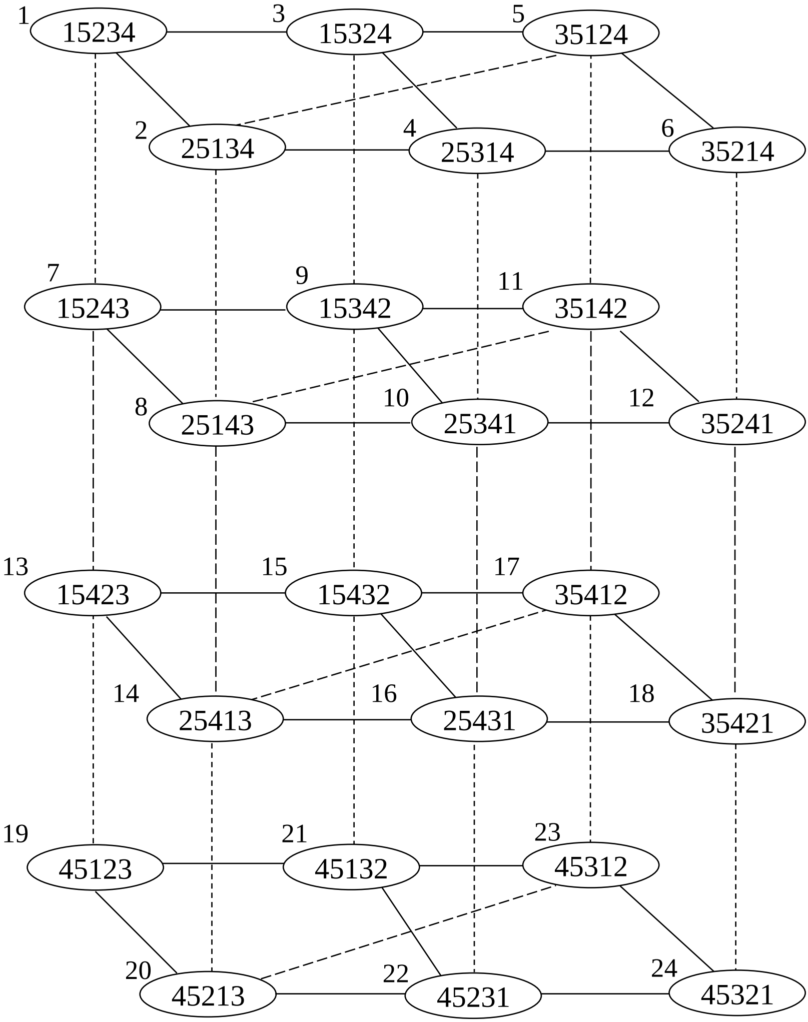


Рис. 3.10. Підпослідовність перестановок
при фіксованому значенні $x_2 = 5$

Далі максимальний елемент переміщується на перше місце і розглядається підграф, в якому вершини визначаються точками перша координата яких $x_1 = 5$.

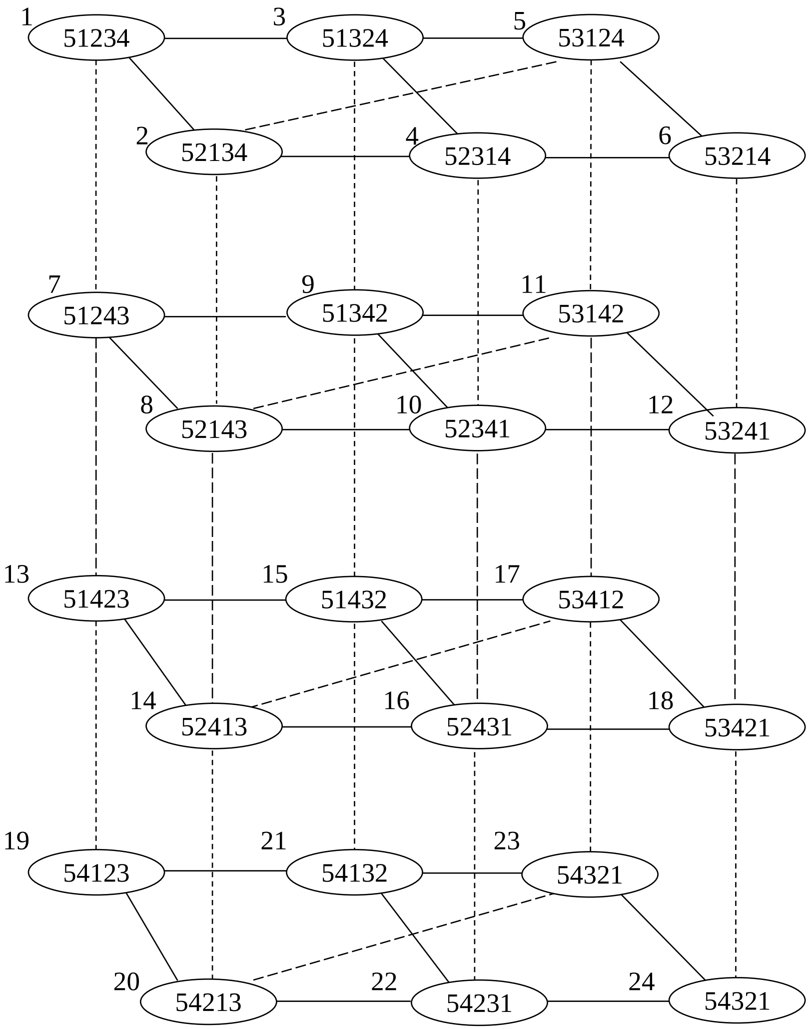


Рис. 3.11. Підпослідовність перестановок
при фіксованому значенні $x_1 = 5$

Розподіл інших елементів координат вершин залишається сталим, як і в інших підграфах, де максимальний елемент знаходиться на іншому місці.

Отже, на основі вище описаного прикладу, можна привести загальний опис рекурентного алгоритму генерування перестановок згідно методу переміщення максимального елементу.

Спочатку для $n = 2$ дві перестановки (12) та (21) як дві вершини з'єднаємо ребром. Переходимо до $n = 3$. Для цього зробимо три копії цього ребра. В першій копії допишемо число 3 на третє місце. На рис.3.4 отримаємо вершини *1* і *2*. В другій копії допишемо число 3 на друге місце, що відповідає вершинам *3* і *4* на рис. 3.4., і з'єднаємо відповідні вершини першої та другої копій. Тепер допишемо число 3 в третій копії на перше місце, що відповідає вершинам *5* і *6* на рис. 3.4, і з'єднаємо відповідні вершини другої та третьої копій.

Тим самим отримали граф перестановок $G_3\left(P\right)$. Далі зрозуміло: якщо побудовано граф перестановок $G_{n-1}\left(P\right)$, то робимо *n* копій цього графа, в *i*-й копії ($1 \le i \le n$) на *i*-му місці в усіх перестановках дописуємо число *n*. Далі з'єднаємо відповідні вершини *i*-ї та $\left(i+1\right)$-ї копій. Тим самим побудовано граф перестановок $G_n\left(P\right)$.

Слід зазначити, що особливістю нових методів генерування комбінаторної конфігурації у вигляді перестановок незалежно від способу генерування є те, що кожний з цих методів дає можливість представити граф, який можна побудувати за деяким способом генерування. В графі вершини розміщені згідно деякої визначеної властивості, тому можна зразу визначити крайні вершини графа, далі граф можна розкласти на підграфи. Для розв'язування екстремальних задач є важливим представлення всієї структури графа з допомогою графів меншої розмірності, або більш простої структури.

Отже, нові методи генерування самі по собі не є важливими, а головним є те, що вони дають можливість користуватися графом, а не зберігати всі об'єкти заданої конфігурації. Надалі є важливим орієнтувати граф та визначити інцидентність вершин та дуг.

Оскільки екстремальні задачі оптимізації розглядаються на різних комбінаторних конфігураціях, то далі розглянемо основні властивості та проаналізуємо методи генерування інших комбінаторних конфігурацій, а також побудову нових методів генерування об'єктів конфігурації і представлення їх в вигляді графів.

3.2.2. Генерування розміщень

Генерація розміщень в загальному випадку має елементарний характер і нараховує безліч методів. Враховуючи те, що для перестановок можна побудувати граф, за допомогою якого можна представити процедуру генерування комбінаторних об'єктів перестановки, розглянемо представлення генерування множини розміщень довжини m з елементами з множини $X = \{1, 2, \ldots, n\}$ у вигляді графа, який позначимо $G_m(A_n)$. При цьому необхідно визначити, які вершини графа рахувати суміжними, тобто з'єднані ребром. Якщо у перестановках бралися до уваги коди суміжних перестановок, які відрізнялися лише однією транспозицією, то у розміщеннях це неможливо, так як вони можуть взагалі не мати спільних елементів. Будемо вважати суміжними два розміщення, які відрізняються або однією транспозицією, або лише одним елементом на одному і тому ж місці. Наведемо приклад генерування розміщень другого типу (за формулою 3.1) для послідовності довжини 3 з елементами з множини $X = \{1, 2, 3, 4\}$. На графі можна представити дані об'єкти в вигляді рис. 3.12.

Дану побудову можна представити також і рекурентним методом. Отже, маємо два числа $n > m \geq 1$ і треба побудувати граф розміщень з $A_n(m) = [n]_m$ вершин. На початку розглянемо випадок $n = m + 1$ (на рис. 3.12). Будуємо граф перестановок для $G_m(P)$, для чого можемо скористатися одним з представлених вище методів. На рис. 3.12 це $G_3(P)$, (вершини 19–24). Далі робимо $n - 1$ копій цього графа. В першій копії (вершини 13–18) робимо скрізь заміну $n - 1$ на n (3 на 4) та з'єднуємо ребрами (пунктирними) відповідні вершини (13, 19), (14, 20) і т. д. Далі по індукції в i-й копії $(n - 1 \geq i \geq 1)$ всюди робимо заміну елементів $n - i$ на $n - i + 1$, потім з'єднуємо ребрами відповідні вершини i-ї та $(i - 1)$-ї копій. Цим і завершується побудова графа розміщень $G_m(A_{m+1})$. Як бачимо, суміжність вершин в цьому графі має двоякий вигляд – в $G_3(P)$ та його копіях, як і в перестановках, завдяки одній транспозиції, а вершин з різних копій – відмінністю тих двох елементів, які замінюються на одному місці. Оскільки для $n = m + 1$ число розміщень дорівнює $m(m-1)(m-2)2 = m!$, то очевидно, що при цьому граф розміщень співпадає з графом перестановок. Загальна схема рекурентної побудови графа розміщень для довільних n та m дещо відрізняється.

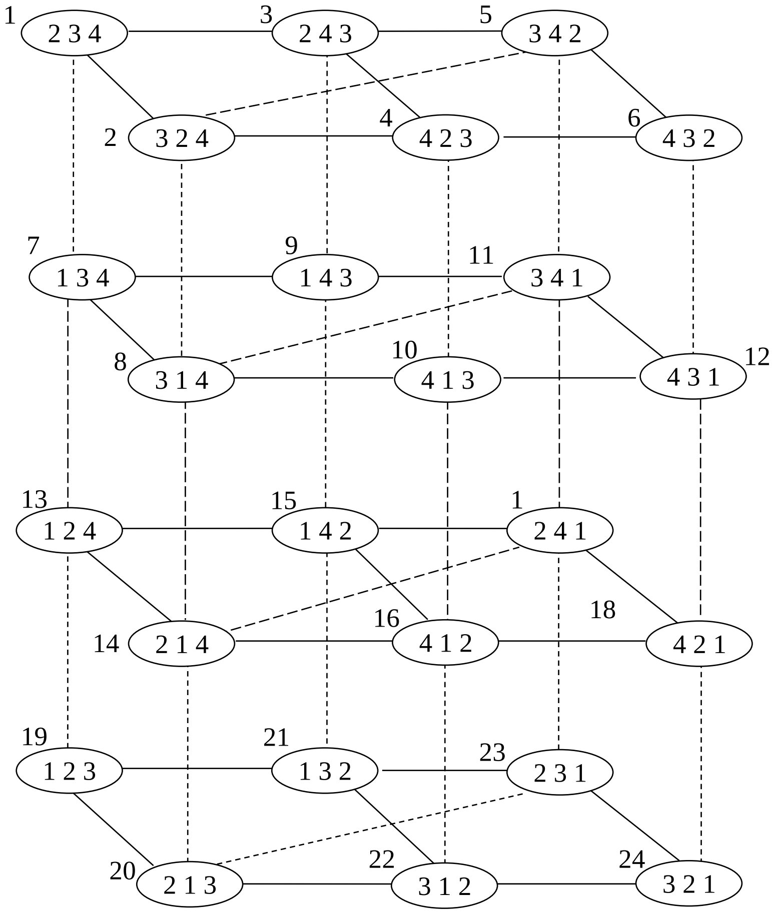


Рис. 3.12. Граф множини розміщень для $n = 4$ та $m = 3$

Почнемо з побудови графа $G_1(A_n)$. Це ізольовані вершини з номерами $1, 2, \ldots, n-1, n$.

Побудова графа $G_m(A_n)$ ведеться по наступній індукції: спочатку будуємо граф $G_1(A_{n-m+1})$. Для цього припускається, що побудовано граф розміщень $G_i(A_{n-m+i})$ $(m \geq i \geq 1)$, де довжина коду розміщень дорівнює i. Далі робимо i +1 копію графа $G_i(A_{n-m+i})$. В j-й копії ($i+1 \geq j \geq 1$) зправа в кожному розміщенні дописуємо число j. При цьому, якщо таке число в роміщенні є зліва, то воно замінюється числом $i+1$. З'єднуємо ребрами відповідні вершини j-ї та $(j-1)$-ї копій ($j>1$). Якщо такі дії виконані для кожного j від 1 до $i+1$, то тим самим побудовано граф $G_{i+1}(A_{n-m+i+1})$.

Коли довжина коду розміщень стане рівною m, то це буде означати, що отримано граф розміщень $G_m(A_n)$. Зазначимо, що відомий граф $G_1(A_3)=1,\ 2,\ 3$.

Нижче наведено таблицю, яка ілюструє дію цього алгоритму для побудови $G_3(A_5)$.

Таблиця 3.1

Побудова підграфів

$\boldsymbol{G_2(A_4)}$	$\boldsymbol{G_3(A_5)}$				
4,1	4,5,1	4,1,2	4,1,3	5,1,4	4,1,5
2,1	2,5,1	5,1,2	2,1,3	2,1,4	2,1,5
3,1	3,5,1	3,1,2	3,1,3	3,1,4	3,1,5
1,2	5,2,1	1,5,2	1,2,3	1,2,4	1,2,5
4,2	4,2,1	4,5,2	4,2,3	5,2,4	4,2,5
3,2	3,2,1	3,5,2	5,2,3	3,2,4	3,2,5
1,3	5,3,1	1,3,2	1,5,3	1,3,4	1,3,5
2,3	2,3,1	5,3,2	2,5,3	2,3,4	2,3,5
4,3	4,3,1	4,3,2	4,5,3	5,3,4	4,3,5
1,4	5,4,1	1,4,2	1,4,3	1,4,4	1,4,5
2,4	2,4,1	5,4,2	2,4,3	2,4,4	2,4,5
3,4	3,4,1	3,4,2	5,4,3	3,4,4	3,4,5

3.2.3. Генерування розбиття множини та числа

Під розбиттям *n*-елементної множини X на k блоків будемо розуміти будь-яке сімейство $\pi = \{B_1 \cup B_2 \cup \ldots \cup B_k\}$, що $B_1 \cup B_2 \cup \ldots \cup B_k = X$, $B_i \cap B_j = \varnothing$ для $1 \le i < j \le k$ та $B_i \ne \varnothing$ для $1 \le i \le k$. Підмножини $B_1, B_2, \ldots, B_k$ будемо називати блоками сімейства π. Множина всіх розбиттів множини X на k блоків будемо позначати через $\Pi(X)$. Очевидно, що $\Pi(X) = \Pi_1(X) \cup \ldots \cup \Pi_n(X)$ (більше того, $\{\Pi_1(X), \ldots, \Pi_n(X)\}$ є розбиттям множини $\Pi(X)$).

Опишемо алгоритм генерування всіх розбиттів множини у вигляді графа. Даний алгоритм легко пояснити, сформулювавши його наступним чином: відмітимо спочатку, що кожне розбиття π множини $\{1, \ldots, n\}$ однозначно визначає розбиття π_{n-1} множини $\{1, \ldots, n-1\}$, що виникає із π після видалення елемента n із відповідного блоку (і видалення утворюваного пустого блоку, якщо елемент n утворював одноелементний блок). Навпаки, якщо дано розбиття $\sigma = \{B_1, \ldots, B_k\}$ множини $\{1, \ldots, n-1\}$, легко знайти всі розбиття π множини $\{1, \ldots, n\}$ такі, що $\pi_{n-1} = \sigma$, тобто наступні розбиття:

$$
\begin{array}{l}
B_1 \bigcup \{n\}, B_2, \ldots, B_k \\
B_1 B_2, \bigcup \{n\}, \ldots, B_k \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
B_1 B_2, \ldots, B_k \bigcup \{n\} \\
B_1 B_2, \ldots, B_k \, \{n\}
\end{array}
\qquad (3.3)
$$

Рис. 3.13. Послідовність розбиттів

Тоді рекурентний метод генерування розбиття множини полягає в наступному: якщо отримано список L_{n-1} всіх розбиттів множини $\{1, \ldots, n-1\}$, то список L_n всіх розбиттів множини $\{1, \ldots, n\}$ будемо одержувати, замінюючи кожне розбиття σ в списку L_{n-1} на відповідну йому послідовність (3.3), згідно рис. 3.14. Зазначимо, що при цьому можна гарантувати, що розбиття, які йдуть один за одним, мало відрізнятиметься одне від одного, точніше кажучи, що кожне наступне розбиття в списку утворюється з попереднього за допомогою видалення деякого елементу з деякого блоку і додавання його в інший блок або створення з нього одноелементного блоку.

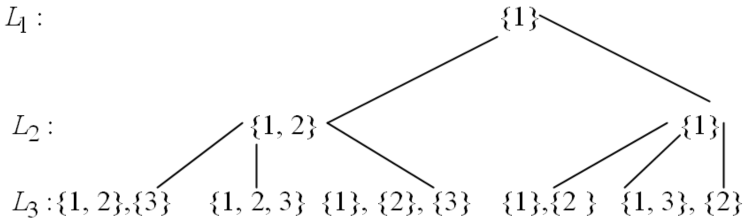


Рис. 3.14. Побудова списків L_1 L_2

Дійсно, послідовні розбиття послідовності (3.3) відповідають цій умові. Якщо порядок послідовності зробити зворотним $\{3, 3\}$ для кожного другого розбиття списку L_{n-1}, то елемент n буде переміщуватися почергово вперед і назад, і розбиття «на стику» послідовностей, утворених із сусідніх розбиттів списку L_{n-1}, будуть мало відрізнятися одне від одного (при умові, що сусідні розбиття списку L_{n-1} мало відрізняються одне від одного). На рис. 3.14 показано побудову списку L_n для множини елементів 1, 2, 3.

Цей спосіб генерування будує спочатку розбиття $\{1,\ldots,n\}$ потім здійснюється переміщення «активного» елементу j у сусідній блок − попередній або подальший (у останньому випадку може виникнути необхідність створення нового блоку елементів, а потім визначення активного елементу в новоутвореному розбитті. З описаної побудови випливає, що даний елемент переміщається тільки тоді, коли всі елементи, що більші нього, досягають свого крайнього лівого або правого положення; точніше, активний елемент j^* є таким найменшим елементом.

Генерування розбиття числа зручно задавати розбиттям $\{p_1, p_2, \dots, p_k\}$ числа n із компонентами, розташованими в деякому порядку, наприклад $p_1 \le p_2 \le \dots \le p_k$.

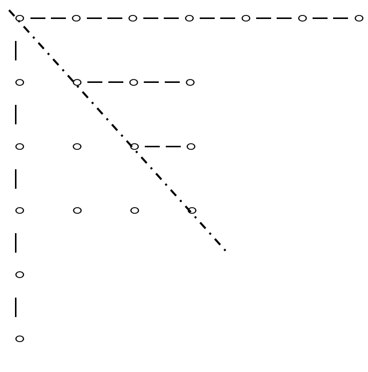
Розбиття n із l компонентами можна породжувати в зростаючому лексикографічному порядку, починаючи із $p_1 = p_2 = \dots = p_{l-1} = 1,\ \ p_l = n - l + 1$ продовжуючи процес таким чином. Для отримання наступного розбиття з поточного переглядаємо елементи справа наліво, зупиняючись на найправішому p_i, такому, що $p_l - p_i \ge 2$. Замінюємо потім p_j на $p_j + 1$ для $j = i,\ i+1, \dots, l-1$ і після цього замінюваний p_l на $n - \sum_{j=1}^{i-1} p_j$.

Іншим ефективним елементарним інструментом вивчення розбиття служить їх графічне представлення. Кожному розбиттю числа ставиться у відповідність його графічне представлення G_λ в вигляді точкового графу. Точковий граф розбиття $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ представляє собою множину цілочислових точок (i, j) на площині, для яких виконується наступна умова

$$(i, j) \in G_\lambda \Leftrightarrow 0 \ge i \ge -n+1,\ 0 \le j \le \lambda_{i+1} - 1,$$

де i – вертикальна вісь, j – горизонтальна вісь.

Точковий граф представляє собою наступну графічну структуру, де частини розбиття розміщуються як правило зверху вниз по спаданню.



3.2.4. Генерування сполучень

Розглянемо методи генерування комбінаторної конфігурації сполучень, тобто генерування *k*-елементних підмножин.

Розглянемо множину $A' = \{1,2,...,n\}$. Сполучення без повторень з *n* елементів по *r* – це *r*-елементна підмножина множини A'. Оскільки порядок запису елементів множини неістотний, то домовимося записувати елементи в кожному сполученні у порядку зростання. Отже, сполучення $\{a_1, a_2, \ldots, a_r\}$, розглядатимемо як рядок чисел $a_1, a_2, \ldots, a_r$, причому $a_1 < a_2 < \ldots < a_r$.

Кількість усіх сполучень без повторень з *n* елементів по *r* позначимо як C_n^r, де *r* і *n* – невід'ємні цілі числа, причому $r \le n$. Кількість усіх сполучень із повтореннями з *n* елементів по *r* позначимо як H_n^r або $H(n,r)$, де *r* і *n* – будь-які невід'ємні цілі числа. Числа C_n^r називають біноміальними коефіцієнтами.

Як і для перестановок, за даним сполученням можна знайти наступне відповідно до лексикографічного порядку. Значення останнього числа в рядку – найбільш можливе, якщо воно дорівнює $n = n - r + r$. Якщо останнє число – найбільш можливе, то передостаннє – найбільш можливе, якщо воно дорівнює $n - r + (r-1)$ або $n - r + i$, де $i = r - 1$ – позиція цього числа. Загалом, значення кожного *i*-го числа найбільш можливе, якщо числа праворуч від нього – найбільші можливі, і це значення дорівнює $n - r + i$. Отже, проглядаємо рядок справа наліво й визначаємо, чи дорівнює значення *i*-го елемента $n - r + i$ (це максимальне значення, яке може бути в *i*-й позиції). Перше значення, яке не задовольняє цю умову, можна збільшити. Нехай, наприклад, це значення дорівнює *m* і займає *j*-ту позицію. Збільшуємо *m* на 1, а значення кожного елемента, що стоїть після *j*-го, дорівнює значенню попереднього елемента плюс 1. Тепер можемо сформулювати потрібний алгоритм.

Алгоритм побудови лексикографічно наступного сполучення полягає у виконанні наступних кроків:

Крок 1. Знайти в рядку $a_1, a_2, \ldots, a_r$ перший справа елемент a_i такий, що $a_i \ne n - r + i$.

Крок 2. Для знайденого елемента виконати присвоювання $a_i := a_i + 1$.

Крок 3. Для $j = i+1, i+2, \ldots, r$, виконати $a_j := a_i + j - i$ (або, що те саме, $\left(a_j := a_{j-1} + 1\right)$).

Даний алгоритм дає можливість будувати наступне в лексикографічному порядку сполучення. Рядок чисел, яким подано лексикографічно наступне сполучення, відрізняється від рядка, що зображає дане сполучення, з позиції *i*, бо в даному сполученні в позиціях $i+1, i+2, \ldots, r$ є максимально можливі числа. Отже, $a_i + 1$ – найменше можливе число, яке можна записати в позицію *i*, якщо хочемо отримати сполучення, більше від даного. Тоді $a_i + 2, \ldots, a_i + r - i + 1$ – найменш можливі числа, які можна записати в позиціях від $i+1$ до *r*.

На основі вище описаного алгоритму можна побудувати послідовність розміщень з *n* елементів по *r*.

Тоді розглядатимемо вище сформульовану задачу лише для множини $A' = \{1, 2, \ldots, n\}$. Один із можливих способів її розв'язання такий. Використаємо алгоритм генерування лексикографічно наступного сполучення для побудови *r*-елементних сполучень *n*-елементної множини A'. Після кожної стадії, коли побудовано чергове *r*-сполучення застосуємо $r! - 1$ разів алгоритм побудови перестановки за умови $n = r$ для побудови всіх перестановок елементів цього сполучення як *r*-елементної множини.

Розглянемо два відомі алгоритми генерування всіх *k*-елементних підмножин *n*-елементної підмножини *X* [148]. Без обмеження загальності можна прийняти $X = \{1, \ldots, n\}$. Тоді кожній *k*-елементній підмножині взаємно однозначно відповідає зростаюча послідовність довжини *k* з елементами з *X:* наприклад, підмножині $\{3, 5, 1\}$ відповідає послідовність $\langle 1, 3, 5 \rangle$.

Розглянемо алгоритм, за допомогою якого генеруються всі такі послідовності в лексикографічному порядку. Достатньо з цією метою зазначити, що при лексикографічному порядку послідовністю, що безпосередньо є наступною за послідовністю $\langle a_1, \ldots, a_n \rangle$, являється послідовність, що визначається наступним чином

$$\langle b_1, \ldots, b_k \rangle = \langle a_1, \ldots, a_{p-1}, a_p + 1, a_p + 2, \ldots, a_p + k - p + 1 \rangle,$$

де $p = max\{i : a_i < n - k + 1\}$.

Більш того, послідовністю, безпосередньо слідуючою за $\langle b_1, \ldots, b_k \rangle$, являється

$$\langle c_1, \ldots, c_k \rangle = \langle b_1, \ldots, b_{p'-1}, b_{p'} + 1, b_{p'} + 2, \ldots, b_{p'} + k - p' + 1 \rangle,$$

де $p' = \begin{cases} p - 1, & якщо \quad b_k = n, \\ k, & якщо \quad b_k < n \end{cases}$

(будемо вважати, що послідовності $\langle a_1, \ldots, a_k \rangle$ і $\langle b_1, \ldots, b_k \rangle$ відмінні від $\langle n - k + 1, \ldots, n \rangle$ останньої послідовності в нашому порядку). Послідовність всіх 4-елементних підмножин множини $\{1, \ldots, 6\}$, отримана за допомогою лексикографічного порядку, можна представити наступним чином рис 3.15.

1	2	3	4
1	2	3	5
1	2	3	6
1	2	4	5
1	2	4	6
1	2	5	6
1	2	4	5
1	3	4	6
1	3	5	6
1	4	5	6
2	3	4	5
2	3	4	6
2	3	5	6
2	4	5	6
3	4	5	6

Рис. 3.15. Послідовність 4-елементних підмножин множини $\{1, \ldots, 6\}$, побудована за допомогою лексикографічного порядку

Другий алгоритм, генерує всі k-елементні підмножини так, що кожна наступна підмножина утворюється з попередньої з видаленням одного елемента і додаванням іншого. Цей алгоритм представимо у рекурсивній формі.

Позначимо з цією метою через $G(n,k)$ список, який має всі k-елементні підмножини множини $\{1,\ldots,n\}$, в якій першою підмножиною є $\{1,\ldots,k\}$, останньою – $\{1,2,\ldots,k-1,n\}$ і кожна наступна підмножина утворюється з попередньої з видаленням деякого елемента і добавлянням другого. Зазначимо, що якщо $G(n-1,k)$ і $G(n-1,k-1)$ вже побудовані, то $G(n,k)$ можна визначити наступним чином:

$$G(n,k)=G(n-1,k-1),G^*(n-1,k-1)\cup\{n\},$$

де $G^*(n-1,k-1)\cup\{n\}$ означає список, утворений з $G(n-1,k-1)$ зміною порядку елементів списку на зворотний та наступним добавляння елемента n до кожної множини. Дійсно $G(n-1,k)$ містить всі k елементні підмножини множини $\{1,\ldots,n\}$, які не містять n, а $G^*(n-1,k-1)\cup\{n\}$ всі k елементні підмножини, які містять n, причому останньою підмножиною в списку $G(n-1,k)$ являється $\{1,2,\ldots,k-1,n-1\}$, а першою підмножиною в списку $G^*(n-1,k-1)\cup\{n\}$ являється $\{1,2,\ldots,k-1,n\}$. На рисунку 3.16 показаний процес побудови списку $G(4,2)$.

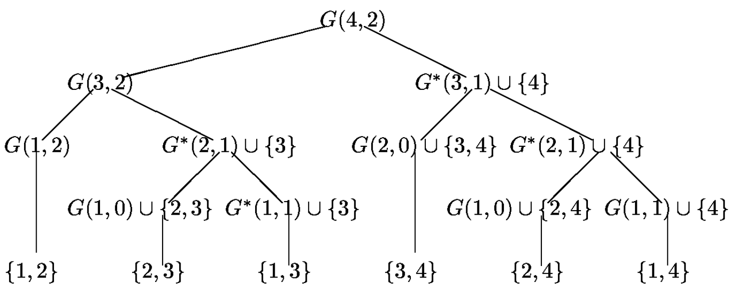


Рис. 3. 16. Побудова списку $G(4,2)$

Списку $G(n,k)$ можемо так як і у випадку генерування всіх підмножин, поставити в відповідність деякий гамільтонів шлях в графі, вершини якого відповідають двоелементним підмножинам множини $\{1,2,3,4\}$, причому вершини, які відповідають підмножинам A і B, з'єднані ребром тоді і тільки тоді, коли $|A \cap B| = k-1 = 1$. Тоді це можна проілюструвати на рис 3.17:

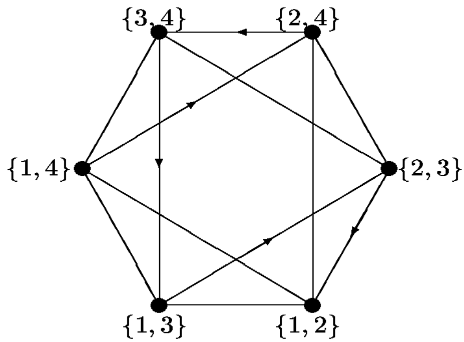


Рис. 3.17. Гамільтонів шлях в графі, що відповідає усім двоелементним підмножинам множини $\{1,2,3,4\}$

Даний список та граф не дає можливості побудувати упорядкування по значенням цільової функції.

Далі розглянемо нові методи генерування комбінаторної конфігурації сполучень, які дають можливість в залежності від складності задачі, представити елементи комбінаторної конфігурації у вигляді графа.

Даний метод генерування полягає у виборі елементів з заданої упорядкованої множини за зростанням, тобто вибираються елементи для прикладу по два перший, другий; перший третій; перший четвертий, і т.д. таким чином будується верхній підграф загального графу послідовності, далі – другий, третій; другий, четвертий, і т. д.

Враховуючи, що кількість усіх сполучень без повторень з n елементів по r дорівнює C_n^r, де r і n – невід'ємні цілі числа, причому $r \le n$, то розглянемо приклади побудови для наступних множин.

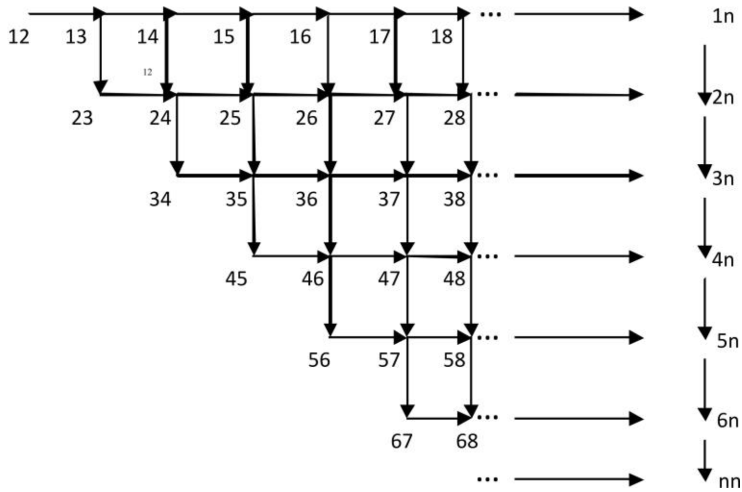


Рис. 3.18. Граф сполучень з *n* по 2

Розглянемо сполучення з 6 по 3, де $C_6^3 = 20$. Формування елементів сполучення здійснюється наступним чином: вибираються з впорядкованої за зростанням множини з шести заданих елементів 1, 2, 3, 4, 5, 6 перші три 1, 2, 3 і для побудови першого підграфу останній елемент замінюється наступними з заданої множини, маємо: 1, 2, 4; 1, 2, 5; 1, 2, 6. Тоді другий підграф утворюється шляхом заміни в кожному з попередніх елементів крім першого другої компоненти на наступний елемент з заданої множини, маємо 1, 3, 4; 1, 3, 5; 1, 3, 6.

Граф на рис. 3.19 можна інтерпретувати як простий тип графів – дерево, оскільки в ньому будь-які дві вершини з’єднані простим ланцюгом.

Дане дерево можна представити у вигляді розкладу на піддерева, тоді кожне піддерево формується з вершин в яких елемент на останньому місці фіксується і є вибраним з початкової множини $a_1, a_2, \ldots, a_n$ як максимальний, а останні перебираються рекурсивним методом.

Слід зазначити, що дане дерево є орієнтованим, в якому визначений корінь – вершина (4 5 6), яка є лексикографічно більша останніх.

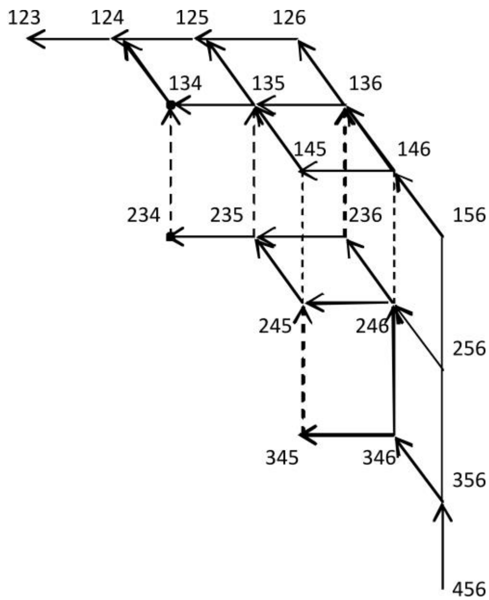


Рис. 3.19. Граф сполучень з 6 по 3, де $C_6^3 = 20$

Тепер можемо сформулювати потрібний алгоритм.

Алгоритм побудови наступного сполучення полягає в виконанні наступних кроків:

Крок 1. Знайти в рядку $a_1, a_2, \ldots, a_r$ перший справа елемент a_i такий, що $a_i \neq n - r + i$.

Крок 2. Для знайденого елемента виконати присвоювання $a_i := a_i + 1$.

Крок 3. Для $j = i+1, i+2, \ldots, r,$ виконати $a_j := a_i + j - i$ (або, що те саме, $\left(a_j := a_{j-1} + 1\right)$.

Даний алгоритм дає можливість будувати наступне в порядку сполучення. Рядок чисел, яким подано лексикографічно наступне сполучення, відрізняється від рядка, що зображає дане сполучення, з позиції *i*, бо в даному сполученні в позиціях $i+1, i+2, \ldots, r$ є максимально можливі числа. Отже, $a_i + 1$ – найменше можливе число, яке можна записати в позицію *i*, якщо хочемо отримати сполучення, більше від даного. Тоді

$a_i + 2, \ldots, a_i + r - i + 1$ – найменш можливі числа, які можна записати в позиціях від $i + 1$ до r.

Як уже було зазначено, відносно інших комбінаторних конфігурацій важливість запропонованого методу генерування комбінаторної конфігурації сполучень в тому, що одержуємо граф, в даному випадку дерево, який може бути використаний для розробки ефективного методу для розв'язування екстремальних задач, оскільки граф відображає всі елементи сполучень.

3.3. Загальна постановка методу направленого структурування

Загальновідомо, що оптимізаційні комбінаторні задачі є одними з найбільш важких з обчислювальної точки зору. Універсальний метод − повний перебір варіантів − застосовний практично для задач малої вимірності. Тому при розв'язанні практичної задачі часто виникає необхідність розробляти нові та удосконалювати існуючі методи, як точні, так і наближені, які враховували б специфіку цільової функції і додаткових обмежень і були б застосовні до задач більшої вимірності, ніж метод повного перебору.

Очевидно, що швидкого поширення набувають алгоритми, які краще і простіше враховують природу класів задач, що розв'язуються. В процесі розробки і реалізації алгоритму природним чином розкриваються властивості, що відображають багатокритеріальність оптимізаційних задач, а також комбінаторні характеристики, які використовувалися в модифікації з новими підходами.

За останні роки в Інституті кібернетики імені В. М. Глушкова НАН України розроблено новий, так званий **метод направленого структурування**, для розв'язування подібних задач.

Основна ідея запропонованого методу перекликається з відомим методом послідовного аналізу варіантів, суть якого описано в першому розділі. Але новий метод призначений безпосередньо для комбінаторних задач. Як відомо, у більшості задач на комбінаторних конфігураціях постає необхідність перераховувати велику кількість варіантів, порівнянну з факторіальною величиною. Тому, метод об'єднує засоби комбінаторного аналізу та теорії графів і передбачає послідовне виконання трьох стадій:

1. Вибір способу генерування у певній послідовності всіх елементів заданої комбінаторної конфігурації, який найбільше пристосований до заданої функції цілі;

2. Представлення множини комбінаторної конфігурації у вигляді орієнтованого графа, де дуга відповідає спаданню значень цільової функції;

3. Побудову поліноміального алгоритму розв'язку задачі на частково упорядкованих вершинах графа.

Завдяки частковій упорядкованості часто вдається трудомісткість алгоритмів звести до логарифма від всієї маси варіантів.

Цей метод було застосовано для розв'язання задач з лінійною та дробово-лінійною цільовими функціями на перестановках, розбиттях, комбінаціях та розміщеннях [56–63]. Створені відповідні інформаційні технології підтвердили ефективність цього методу. Продовжуються дослідження з метою створення відповідних алгоритмів для розв'язання комбінаторних задач з іншими цільовими функціями та на інших комбінаторних конфігураціях.

Висновки до розділу

В розділі зроблена порівняльна характеристика існуючих методів генерування комбінаторних конфігурацій – лексикографічного, антилексикографічного та розглянуто метод генерування всіх перестановок за мінімальне число транспозицій. Побудовано та описано нові методи генерування різних комбінаторних конфігурацій – рекурсивний метод та метод переміщення максимального елемента. В залежності від властивостей задачі визначається той чи інший метод генерування.

Нові методи генерування дають можливість представити множину комбінаторних конфігурацій у вигляді графа комбінаторних конфігурацій, який відображає елементи конфігурації і по якому можна рухатися в залежності від значення функції. Граф комбінаторної конфігурації в подальшому буде застосовано для побудови методів розв'язання екстремальних комбінаторних задач.

На основі графа генерування запропоновано та описано схему нового методу направленого структурування.

Отже, враховуючи вищезазначене, для розв'язування складних екстремальних задач було розроблено новий, так званий метод направленого структурування. Його основна ідея перекликається з відомим методом послідовного аналізу варіантів, але спеціально пристосованого для розв'язування комбінаторних задач. Як відомо, у більшості задач на комбінаторних конфігураціях постає необхідність перераховувати велику кількість варіантів, порівняну з факторіальною величиною. Метод об'єднує засоби комбінаторного аналізу та теорії графів і дає можливість обійти великий перебір елементів конфігурації, на якій розглядається задача.

РОЗДІЛ 4. ОПТИМІЗАЦІЯ ЛІНІЙНОЇ ФУНКЦІЇ НА ПЕРЕСТАНОВКАХ

Багато екстремальних задач комбінаторної оптимізації, такі як планування роботи підприємства, розподіл ресурсів, задача управління, мережеве планування описуються моделями дискретної оптимізації. Із задач дискретного моделювання виділяються задачі комбінаторної оптимізації, які виникають в найрізноманітніших галузях людської діяльності [8, 11, 46–50, 52, 140, 145, 147–165, 238–240].

Задачі на комбінаторних конфігураціях цікаві тим, що область допустимих розв'язків є деяким комбінаторним многогранником, властивості якого вивчені і досліджені. Знання специфічних властивостей комбінаторного многогранника дає можливість використовувати їх для побудови нових і для вдосконалення існуючих методів розв'язання комбінаторних оптимізаційних задач.

При вивченні графів многогранників виникає велика кількість задач, що представляє інтерес не тільки для теорії графів, комбінаторики, топології, але і для теорії лінійного програмування. Використання властивостей графів комбінаторних многогранників можуть послужити підвищенню ефективності «традиційних» і розробці нових методів комбінаторної оптимізації. Комбінаторні моделі можуть бути застосовані для представлення оптимізаційних задач, що виникають при оптимальному розміщенні на графах. Комбінаторна теорія многогранників вивчає екстремальні властивості многогранників, розглядаючи множину його граней всіх розмірностей як деякий комплекс. Але при розв'язанні таких задач виникають проблеми, пов'язані з складністю математичних моделей, великим об'ємом інформації і т. д., оскільки більшість задач на комбінаторних множинах є *NP*-повними. Більшість задач на графах стосується визначення компонент зв'язності, пошуку маршрутів, відстаней і т. п. Проте при розв'язанні прикладних задач відповідні їм графи досить великі, а аналіз можливий лише із залученням сучасної обчислювальної техніки. Отже, метою даного розділу є опис і реалізація нового підходу для розв'язування екстремальних задач з лінійною функцією на комбінаторних конфігураціях з застосуванням теорії графів та многогранників. Зокрема, в розділі досліджується комбінаторна задача на різних комбінаторних конфігураціях перестановок, сполучень, розміщень, що представлені в [88, 90–93, 96–100, 239, 242–245]. На підставі встановленого взаємозв'язку між задачами на комбінаторних конфігураціях і їх графами вивчаються деякі структурні властивості допустимої області, а також сформульовано ряд тверджень, що дають можливість застосувати для побудови методів розв'язання екстремальних задач з використанням графів. Зокрема, в роботі [58] описаний спосіб побудови гамільтонового шляху усередині гіпергранeй. У статті [61] ставиться задача про побудову гамільтонового шляху між гіпергранями, тобто, як добудувати гамільтонів шлях, помістивши між двома підграфами третій підграф, якщо гамільтонів шлях усередині двох перших підграфів побудовано.

4.1. Загальна математична постановка задачі на комбінаторних конфігураціях

Загальна задача комбінаторної оптимізації полягає у відшуканні екстремуму довільної цільової функції на довільних комбінаторних конфігураціях. Як правило, при розв'язанні класу таких задач досліджується можливість їх лінеаризації, тобто побудови опуклої оболонки допустимих розв'язків задачі. Перехід від параметричної форми задання опуклого многогранника до аналітичної має велике значення для задач дискретної оптимізації, оскільки дозволяє сформулювати їх в термінах лінійного програмування. Але застосування методів лінійного програмування не завжди відповідає характеру екстремальних задач. Застосування теорії графів до розв'язання таких задач дає в багатьох випадках можливість отримати розв'язок з меншими витратами комп'ютерних ресурсів. Підзадачею вище сформульованої задачі може бути визначення гамільтонового шляху, який визначає зміну значення цільової функції на множині комбінаторних конфігурацій. Многогранник гамільтонових циклів графа є гранню многогранника гамільтонових циклів повного графа. Отже, розглянемо побудову послідовності значень лінійної цільової функції на графі перестановок. При цьому не обов'язково будувати повний граф, достатньо знайти таку часткову упорядкованість, яка дозволить виділити такі спадаючі (або зростаючі) за значеннями функції шляхи, які приведуть нас до екстремуму функції. Це зобов'язує більш детально вивчати цільову функцію та враховувати властивості і структуру послідовності перестановок, на якій розглядається задача.

Розглянемо екстремальну задачу комбінаторної оптимізації вигляду:

$$Z\left(\Phi,P\left(A\right)\right):max\left\{\Phi\left(a\right)\mid a\in P\left(A\right)\right\},$$

яка полягає в максимізації функції $\Phi\left(a\right)$ на множині перестановок $P\left(A\right)$, де $\Phi\left(a\right)=\sum_{j=1}^{n}c_jx_j$. Відомо, що можна сформулювати екстремальну задачу $Z\left(F,X\right)$ максимізації критерію $F\left(x\right)$ на множині X, причому кожній точці $a\in P_{nk}\left(A\right)$ відповідатиме точка $x\in X$, така, що $F\left(x\right)=\Phi\left(a\right)$.

$$Z\left(F,X\right):max\left\{F\left(x\right)\mid x\in X\right\}, \tag{4.1}$$

де $F\left(x\right)=\sum_{j=1}^{n}c_jx_j$, X – не порожня множина в R^n, яка визначається таким чином $X=vert\,\Pi\left(A\right)$, $\Pi=conv\,P\left(A\right)$.

Слід зазначити, що іноді є доцільним розв'язувати задачу вигляду:

$$x^*=\underset{x\in\Pi\left(A\right)}{arg\,max}\,F\left(x\right),$$

для значення функції $y^*=F\left(x^*\right)$. Так само має сенс розглядати різновид вище зазначеної задачі, де значення цільової функції знаходиться в інтервалі

$$F\left(\bar{x}\right)\le F\left(x\right)\le F\left(\bar{\bar{x}}\right). \tag*{(4.1)$'$}$$

Тоді задача прийме вигляд: визначити

$$\bar{x} = \underset{x \in \Pi(A)}{arg\,max}\, F(x) \text{ при } \bar{y} = F(\bar{x}),$$

$$\bar{\bar{x}} = \underset{x \in \Pi(A)}{arg\,max}\, F(x) \text{ при } \bar{\bar{y}} = F(\bar{\bar{x}}) \qquad (4.2)$$

при умові $\left|\bar{x} - \bar{\bar{x}}\right| \to min$.

Розглядаємо перестановку як впорядковану вибірку елементів $a = \left(a_{i_1}, a_{i_2}, \ldots, a_{i_n}\right)$, де $a_{i_j} \in A\ \forall i_j \in N_n$, $\forall j \in N_n$, $i_s \neq i_t$, якщо $s \neq t\ \forall s \in N_n$, $\forall t \in N_n$ з деякої мультимножини $A = \left\{a_1, a_2, \ldots, a_q\right\}$, яке характеризується основою $S(A) = \left\{e_1, e_2, \ldots, e_k\right\}$, де $e_j \in R^1, \forall j \in N_k$ і кратністю елементів $k\left(e_j\right) = r_j$, $j \in N_k$, $r_1 + r_2 + \ldots + r_k = q$ згідно [98, 242, 245].

Множина перестановок з повтореннями з *n* дійсних чисел, серед яких *k* різних, називається загальною множиною перестановок і позначається $P_{nk}(A)$. Це множина впорядкованих *n*-вибірок з мультимножини *A* при умові $n = q > k$.

При $n = k = q$ маємо множину перестановок без повторень. Позначимо її P_n. Очевидно, що $P_n(A) = P_{nn}(A)$. У тих випадках, коли не указується вигляд множині перестановок, записуватимемо ці множини, як $P(A)$. Відомо [87, 98], що опуклою оболонкою множини перестановок є переставний многогранник $\Pi(A) = conv\,P(A)$, множина вершин якого рівна множині $P(A)$ перестановок: $vert\,\Pi(A) = P(A)$.

Не втрачаючи загальності, упорядкуємо елементи мультимножини *A* по неспаданню:

$$a_1 \leq a_2 \leq \ldots \leq a_n, \qquad (4.3)$$

і елементи його основи – по зростанню: $e_1 < e_2 < \ldots < e_k$. Тоді опуклою оболонкою загальної множини перестановок $P(A)$ є загальний переставний многогранник $\Pi(A) = conv\,P(A)$, який описується відомою системою лінійних нерівностей:

$$\begin{cases} \sum_{j=1}^{n} x_j \le \sum_{j=1}^{n} a_j, & (4.4) \\ \sum_{j=1}^{i} x_{\alpha_j} \ge \sum_{j=1}^{i} a_j, & (4.5) \end{cases}$$

$\alpha_j \in N_n, \alpha_j \ne \alpha_t, \forall j \ne t, \forall j,t \in N_i, \forall i \in N_n$, а $P(A) = vert\,\Pi(A)$.

Розглянемо підхід до розв’язання задач, що ґрунтується на впорядковуванні значень цільової лінійної функції $F(x)$ і побудові гамільтонового шляху для точок, в яких ці значення досягаються, а потім в застосуванні методу дихотомії до визначеного гамільтонового шляху. Далі під задачею $Z(\Phi, P(A))$ розуміємо задачу $Z(F,X)$.

Для побудови методу, перш за все, на початковому етапі необхідно визначити початкову точку. Розглянемо факт у вигляді твердження.

Твердження 4.1. Якщо для елементів мультимножини A і коефіцієнтів цільової функції задачі

$$extr\left\{ f(x) = \sum_{j=1}^{n} c_j x_j \mid x \in vert\,\Pi(A) \right\} \tag{4.6}$$

виконуються відповідно умови (4.3) і

$$c_{i_1} \le c_{i_2} \le \ldots \le c_{i_n}, \tag{4.7}$$

$i_n \in N_n$, то максимум функції $f(x)$ на допустимій множині досягається в точці $x^* = \left(x^*_{i_1}, \ldots, x^*_{i_n}\right) \in vert\,\Pi(A)$, яка задається таким чином:

$$x^*_{i_j} = a_j \ \forall\ j \in N_n,$$

а мінімум відповідно в точці $y = (y_1, y_2, \ldots, y_n)$, де

$$y_{i_{j+1}} = a_{n-j} \ \forall\ j \in N_{n-1} \cup \{0\}.$$

Слід зазначити, що загальне число q лінійних нерівностей, що входять в систему (4.4), (4.5), що описує переставний многогранник $\Pi(A)$ рівно $\sum_{i=0}^{n} C_n^i = 2^n$, а це задача великої розмірності і є дуже складною при розв'язанні традиційними методами лінійного програмування. Тому є необхідність в розробці нових методів, які базуються на властивостях множини допустимих розв'язків і цільових функцій.

Для даної задачі (4.6) область допустимих розв'язків визначає переставний многогранник, вершини якого є точками загальної множини перестановок.

Як відомо з попереднього розділу, множину перестановок можна представити у вигляді графа перестановок $G(P_n)$, який є об'єднанням підграфів.

Тоді можна стверджувати, що вершини переставного многогранника, які є точками загальної множини перестановок $P(A)$, розміщені у вершинах графа перестановок $G(P_n)$.

Граф перестановок має ряд цікавих властивостей, тому використання властивостей графа множини перестановок дають можливість при реалізації методу для розв'язання екстремальної задачі $Z(F,X)$ мінімізувати витрати часу на перевірку належності знайденої точки комбінаторним обмеженням многогранника і зменшити кількість обмежень в початковій системі.

4.2. Метод упорядкування значень лінійної функції на перестановках

Як було визначено вище, розглядається задача, в якій необхідно визначити точку – вершину переставного многогранника $\Pi(A)$ за відомим значенням цільової функції. Для цього спочатку достатньо знайти значення цільової функції в кожній точці, побудувати для цих значень шлях (орієнтований ланцюг), який відображає монотонність цільової функції, переходи від точки до точки, і з'ясувати залежність між ними.

Для генерації всіх $n!$ перестановок n – елементної множини існує багато методів, огляд яких зроблено в другому розділі. Використовуючи описані деякі методи генерування комбінаторних множин, генеруємо послідовність перестановок і формуємо графічні структури.

Послідовносі перестановок $P(A)$, визначаємо як вершини графа $G(P_n)$, що відповідають всім точкам множини перестановок. Для такого графа дві вершини будемо визнавати суміжними, якщо коди відповідних перестановок відрізняються одноразовою транспозицією двох елементів. З'єднуючи відповідні вершини по ходу генерації перестановок, одержуємо спочатку неорієнтований граф. Немає необхідності (як можна впевнитися далі) з'єднувати всі суміжні вершини, а тільки ті, що генеруються в даний момент і будь-який гамільтонів ланцюг відповідає деякому варіанту генерування всіх перестановок. Якщо в кожній вершині графа знайти відповідне значення заданої цільової функції $F(x)$, то відносно значень функції можна побудувати орієнтований граф, зорієнтувавши його ребра у напрямі від більшого значення функції у суміжних вершинах до меншого. Досить актуальною для розробки нових методів розв'язування задачі є питання побудови гамільтонового шляху, уздовж якого всі значення функції строго впорядковані за спаданням (збільшенням). Очевидно, що кінцеві точки гамільтонового шляху визначають екстремальні точки (перестановки). Тоді є важливим наступний факт, який можна сформулювати у вигляді леми.

Лема 4.1. Якщо з перестановки $p_1 = (i_1, i_2, \ldots, i_n) \in P(A)$ одержана перестановка $p_2 \in P(A)$ транспозицією двох чисел $i_k < i_l$, де $k < l$, то $F(p_1) \geq F(p_2)$.

Розглянемо лінійну функцію

$$F(x,c) = (c,x) = \sum_{i=1}^{n} c_i x_i. \tag{4.8}$$

Тут $c = (c_1, c_2, \ldots, c_n)$ множина довільних чисел, що визначають коефіцієнти цільової функції, а $x = (x_1, x_2, \ldots, x_n)$ – перестановка чисел $(1, 2, \ldots, n)$. Розглянемо *u*-перестановку з симетричної групи перестановок S_n. Нехай

$$F\left(x,c,u\right)=\sum_{i=1}^{n}c_i x_{u(i)}. \tag{4.9}$$

Основна проблема комбінаторної оптимізації полягає в тому, щоб знайти таку перестановку u_0, що $F\left(x,c,u_0\right)\le F\left(x,c,u\right)$ для довільного $u\in H,$ де H – довільна не порожня підмножина симетричної групи S_n. Якщо $t\in S_n$, то позначимо $c^t=\left(c_{t(1)},c_{t(2)},\ldots,c_{t(n)}\right),$ $x^t=\left(x_{t(1)},x_{t(2)},\ldots,x_{t(n)}\right)$.

Тоді $F\left(x,c\right)=F\left(c^t,x^t\right)=\left(c,x\right)=\left(c^t,x^t\right)$. Аналогічно

$$F\left(x,c,u\right)=F\left(x^t,c^t,u'\right),$$

де $u'=t^{-1}ut$ для всякого u з S_n. Отже, при розв’язанні основної проблеми завжди можна замінити пару $\left(x,c\right)$ на пару $\left(x^t,c^t\right)$. Зокрема, завжди можна вважати:

$$c_1\le c_2\le\ldots\le c_n, \tag{а}$$

або

$$c_1\ge c_2\ge\ldots\ge c_n. \tag{б}$$

Якщо $H=S_n$, то проблема розв’язана [57]. Доведено, що якщо має місце (а), то максимум функції $F\left(x,c,u\right)$ досягається для перестановки $u_0=\left(1,2,\ldots,n\right),$ а мінімум – для перестановки $u^*=\left(n,n-1,\ldots,2,1\right)$. У подальших дослідженнях передбачатиметься, що для коефіцієнтів c_i $\left(1\le i\le n\right)$ завжди має місце (а). Розглянемо тепер детально структуру відповідних графів для невеликих значень n. Для $n=3$ граф зображений на рис. 4.1, де дуга, що виходить з перестановки p_3 і заходить в перестановку p_j рівносильна співвідношенню $F\left(x,c,p_1\right)\ge F\left(x,c,p_2\right)$.

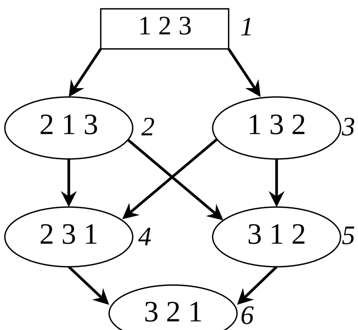


Рис. 4.1. Структура значень $F(x)$ для $n=3$

Зазначимо, що на рис. 4.1. дуги відображають зміну значень цільової функції, а вершини графа побудовано згідно рекурсивного методу, що описаний в другому розділі. Доведемо, що граф однозначно відображає впорядкування значень функцій. Для цього достатньо зробити безпосереднє обчислення перестановок.

Наприклад: $-\begin{array}{ccc} 1 & 3 & 2 \\ 1 & 2 & 3 \\ \hline 0 & 1 & -1 \end{array}$. Різниця відповідає різниці значень функції $c_2 - c_3 \le 0,$ що відповідає дійсності. На цьому графі немає з'єднань між перестановками $(1\ 3\ 2)$ і $(2\ 1\ 3)$, а так само між $(2\ 3\ 1)$ і $(3\ 1\ 2)$. Цим парам відповідає різниця $(1 - 2\ 1)$, що рівносильна $c_1 - 2c_2 + c_3,$ значення якої не може бути визначено однозначно.

Розглянемо тепер граф для $n=4,$ що вже побудовано.

Приклад 4.1. Нехай, дана множина $A=\{1,2,3,4\}$, за допомогою якої утворюється множина перестановок $P(A)$. Визначена функція $F(x)=c_1x_1+c_2x_2+c_3x_3+c_4x_4$, коефіцієнти якої впорядковані таким чином $c_1 \le c_2 \le c_3 \le c_4$ і приймають значення $\{1,2,3,4\}$. Зазначимо, що якщо коефіцієнти цільової функції є невпорядковані, то згідно з визначенням 1.1. нормалізуємо функцію $F(x)=\sum_{i=1}^{n} c_i x_i$, за допомогою відображення перестановки $\varphi: N \to C,$ що встановлює впорядкування коефіцієнтів

$c_1, c_2, \ldots, c_n$ цільової функції по зростанню. Тоді функція є нормалізованою і відповідає умові задачі. Нехай задано значення $y^* = F\left(x^*\right)$ функції в деякій точці.

Необхідно знайти $x^* = arg\, extrF\left(x\right)$, де $x^* \in P\left(A\right)$.

Розв'язання. Представимо розкладання графа переставного многогранника $\Pi\left(A\right)$ на підграфи, в яких виділено зафіксований елемент для функції, згідно рекурсивного методу, описаного в другому розділі. Згідно умові впорядкування коефіцієнтів, у прямокутній вершині підграфа досягається максимальне значення цільової функції. Зафіксуємо ті перестановки, для яких $x_4 = 3$. Оскільки при обчисленні різниці кодів на четвертій позиції завжди буде 0, то очевидно, що цей підграф буде точною копією графа, зображеного на рис. 4.1. Підграф при $x_4 = 4$ представлений на рис. 4.2а якщо об'єднати перестановки, в яких на четвертому місці стоїть цифра 3, то одержимо підграф, представлений на рис. 4.2б. Аналогічно будуть представлені графи перестановок, де на останньому місці елементів зафіксовані цифри 2 і 1 відповідно.

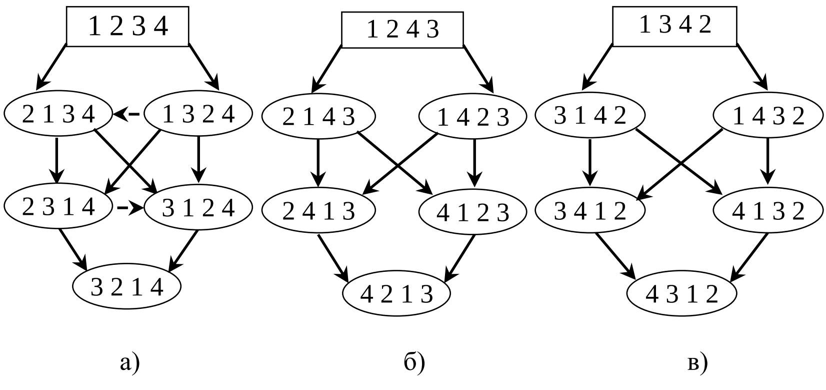


Рис. 4.2. Розкладання графа переставновок на підграфи

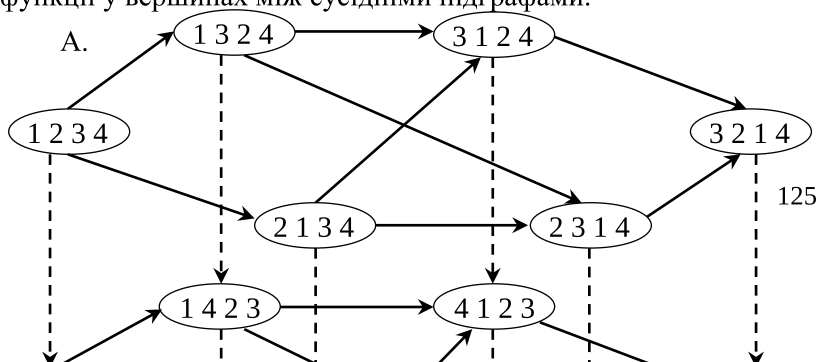
На рис. 4.2. стрілки указують перехід від точки до точки за спаданням значень цільових функцій, згідно теореми. Слід зазначити, що практично всі зв'язки визначені, але між сусідніми елементами всередині як і на рис. 4.2, зв'язок необхідно дослідити і вивчити, тоді можна буде представити повний гамільтонів шлях на графі переставного многогранника.

Позначимо перший підграф − *A*, що зображено на рис. 4.2а, то, можна зобразити всі підграфи, які одержані з *A* за допомогою однієї транспозиції. Слід зазначити, що у всіх графах однозначно визначається максимальне і мінімальне значення функції. І взагалі, всі графи є копією графа *A*, так що їх можна в тому ж порядку помістити один під одним, об'єднавши в загальний граф послідовності перестановок і вказавши зміну значень заданої цільової функції в даних перестановках в порядку спадання. Неважко помітити, що граф *B* отримаємо з графа *A*, якщо в останньому у всіх перестановках зробити транспозицію чисел 4 і 4. Аналогічно граф *C* отримаємо з графа *B*, якщо в останньому зробити транспозицію $(3,\ 2)$, а граф *D* з графа *C* після транспозиції чисел $(2,\ 1)$. Звідси випливає, що значення функції тим більше, чим вище знаходиться відповідний підграф. Це можна представити схематично так

$$F(x)_{[A]} \geq F(x)_{[B]} \geq F(x)_{[C]} \geq F(x)_{[D]}. \qquad (4.10)$$

Об'єднання всіх підграфів дає загальний орієнтований граф $G(P_n) = A \bigcup B \bigcup C \bigcup D.$ На рис. 4.3 представлений граф для $n = 4,$ який є об'єднанням підграфів. Причому підграфи є ізоморфними, а отже еквівалентними, оскільки ізоморфізм графів є відношення еквівалентності.

Аналізуючи рис. 4.1, 4.2, 4.3, приходимо до висновку, що в кожному підграфі A, B, C, D залишаються по дві незрівнянні вершини. Для знаходження гамільтонового шляху із спадаючими значеннями функції необхідно порівняти значення функції в цих вершинах, крім того ще потрібно порівняти значення функції у вершинах між сусідніми підграфами.



B.

C.

D.

Рис. 4.3. Граф послідовності перестановок

На підставі вище висловлених міркувань і рис. 4.1, 4.2, 4.3 можна зробити висновок, що точки множини перестановок $P(A)$ можна розкласти за підграфами, або паралельними гіперплощинами у порядку спадання значень лінійної цільової функції $F(x)$ у цих точках.

Розкладання точок комбінаторної множини перестановок $P(A)$ при $n \geq 4$ забезпечує ієрархічне розташування цих точок по гіперплощинах *A, B, C, D,* (рис. 4.2) згідно значень цільової функції $y^* = F\left(x^*\right)$.

Введемо наступні позначення $\Delta_1 = c_2 - c_1;$ $\Delta_2 = c_3 - c_2;$ $\Delta_3 = c_4 - c_3,$ для подальшого розгляду властивостей графа рис. 4.4. Між ними можуть бути різні співвідношення залежно від їх конкретних значень. Для вище введених позначень встановимо можливі співвідношення:

1) $\Delta_1 = \Delta_2 = \Delta_3;$ це коли всі різниці однакові. Якщо вони різні, то в залежності від величин можливі шість співвідношень

$$\begin{aligned} &1\,2\,3 \to 2)\,\Delta_1 > \Delta_2 > \Delta_3;\; 1\,2\,3 \to 3)\,\Delta_1 > \Delta_3 > \Delta_2;\\ &2\,1\,3 \to 4)\,\Delta_2 > \Delta_1 > \Delta_3;\; 2\,3\,1 \to 5)\,\Delta_2 > \Delta_3 > \Delta_1;\\ &3\,1\,2 \to 6)\,\Delta_3 > \Delta_1 > \Delta_2;\; 3\,2\,1 \to 7)\,\Delta_3 > \Delta_2 > \Delta_1. \end{aligned} \tag{4.11}$$

Можна розглянути ще і окремі випадки:

1) $\Delta_1 = \Delta_2 > \Delta_3;$ 2) $\Delta_1 = \Delta_2 < \Delta_3;$ 3) $\Delta_1 = \Delta_3 > \Delta_2;$ 4) $\Delta_1 = \Delta_3 < \Delta_2;$ 5) $\Delta_2 = \Delta_3 > \Delta_1;$ 6) $\Delta_2 = \Delta_3 < \Delta_1.$

Для побудови гамільтонового шляху необхідно встановити співвідношення на кожному з підграфів *A, B, C, D,* (рис. 4.2) між парами точок (3;2), (5;4) (точки пронумеровані на рис. 4.4).

По кожному з цих випадків необхідно обчислити схему на підграфах *A, B, C, D,* потім скласти загальне співвідношення і вказати гамільтонів шлях по всьому переставному многограннику $\Pi(A)$. Як приклад розглянемо перший випадок, коли $\Delta_1 = \Delta_2 = \Delta_3.$

Верхній підграф *A* одержує дві пари вершин (3;2), (5;4), де значення функції співпадає і між якими необхідно встановити зв'язок. Тому обчислимо співвідношення:

$$\begin{array}{cccc} 1 & 3 & 2 & 4 \\ -2 & 1 & 3 & 4 \\ \hline -1 & +2 & -1 & \end{array} = \Delta_1 - \Delta_2 = 0, \quad \begin{array}{cccc} 3 & 1 & 2 & 4 \\ -2 & 3 & 1 & 4 \\ \hline +1 & -2 & +1 & \end{array} = -\Delta_1 + \Delta_2 = 0.$$

Тоді, можна зобразити наступну схему для гіперплощини *A*:

$A.$

$$1\,2\,3\,4 \rightarrow \begin{matrix} 1\,3\,2\,4 \\ 2\,1\,3\,4 \end{matrix} \rightarrow \begin{matrix} 3\,1\,2\,4 \\ 2\,3\,1\,4 \end{matrix} \rightarrow 3\,2\,1\,4$$

Аналогічна ситуація в підграфі *D*:

$D.$

$$2\,3\,4\,1 \rightarrow \begin{matrix} 2\,4\,3\,1 \\ 3\,2\,4\,1 \end{matrix} \rightarrow \begin{matrix} 4\,2\,3\,1 \\ 3\,4\,2\,1 \end{matrix} \rightarrow 4\,3\,2\,1$$

Якщо розглянути підграф *B* (рис. 4.2), то одержуємо співвідношення для таких пар вершин

$$\begin{array}{cccc} 1 & 4 & 2 & 3 \\ -2 & 1 & 4 & 3 \\ \hline -1 & +3 & -2 & \end{array} = \Delta_1 - 2\Delta_2 = -\Delta_2 < 0$$

$$-\begin{array}{cccc} 4 & 1 & 2 & 3 \\ 2 & 4 & 1 & 3 \\ \hline +2 & -3 & +1 & \end{array} = -2\Delta_1 + \Delta_2 < 0$$

На підграфі *B* виникають співвідношення $\Delta_1 - 2\Delta_2$ і $-2\Delta_1 + \Delta_2$. Тому гамільтонів шлях на *B* має вигляд:

B.

$$1\,2\,4\,3 \rightarrow 2\,1\,4\,3 \rightarrow 1\,4\,2\,3 \rightarrow 2\,4\,1\,3 \rightarrow 4\,1\,2\,3 \rightarrow 4\,2\,1\,3$$

На підграфі *C* аналогічна ситуація: $1\ 4\ 3\ 2 > 3\ 1\ 4\ 2$ і $4\ 1\ 3\ 2 > 3\ 4\ 1\ 2$, тому маємо:

C.

$$1\,3\,4\,2 \rightarrow 1\,4\,3\,2 \rightarrow 3\,1\,4\,2 \rightarrow 4\,1\,3\,2 \rightarrow 3\,4\,1\,2 \rightarrow 4\,3\,1\,2$$

Введемо в розгляд поняття α_i-питань, що необхідно дослідити для взаємовідношення між внутрішніми точками підграфів, що розглядаються на рис. 4.2.

При розв’язанні α_i -питань, також слід зазначити, що простежується залежність між точками, які знаходяться на різних гіперплощинах.

Відзначимо, що на гіперплощинах *A* і *B* є точки, для яких виконується наступне співвідношення:

$$\begin{array}{cccc} 1 & 3 & 2 & 4 \\ -1 & 2 & 4 & 3 \\ \hline +1 & -2 & +1 & \end{array} = -\Delta_3 + \Delta_4 = 0$$ тобто їх значення рівні.

Аналогічно, значення в точці $(3\ 1\ 2\ 4)$ на гіперплощині *A* дорівнює значенню в точці $(1\ 4\ 2\ 3)$ на гіперплощині *B*, оскільки

$$\begin{array}{cccc} 3 & 1 & 2 & 4 \\ -1 & 4 & 2 & 3 \\ \hline +2 & -3 & 0 & 1 \end{array} = -2\Delta_1 + \Delta_2 + \Delta_3 = 0,$$

а значення в точці $(3\ 2\ 1\ 4)$ з гіперплощини *A* більше, ніж значення в точці $2\ 4\ 1\ 3$ з гіперплощини *B*, оскільки виконується умова:

$$\begin{array}{cccc} 3 & 2 & 1 & 4 \\ -2 & 4 & 1 & 3 \\ \hline +1 & -2 & 0 & 1 \end{array} = -\Delta_1 + \Delta_2 + \Delta_3 > 0.$$

Розрахуємо значення цільової функції в точках:

$$F(x_1) = 1\cdot 1 + 2\cdot 2 + 3\cdot 3 + 4\cdot 4 = 30$$

$$F(x_7) = 1\cdot 2 + 2\cdot 3 + 3\cdot 1 + 4\cdot 4 = 27$$

$$F(x_2) = 1\cdot 1 + 2\cdot 3 + 3\cdot 2 + 4\cdot 4 = 29$$

$$F(x_8) = 1\cdot 1 + 2\cdot 4 + 3\cdot 2 + 4\cdot 3 = 27$$

$$F(x_3) = 1\cdot 2 + 2\cdot 1 + 3\cdot 3 + 4\cdot 4 = 29$$

$$F(x_9) = 1\cdot 3 + 2\cdot 2 + 3\cdot 1 + 4\cdot 4 = 26$$

$$F(x_4) = 1\cdot1 + 2\cdot2 + 3\cdot4 + 4\cdot3 = 29$$

$$F(x_{10}) = 1\cdot2 + 2\cdot4 + 3\cdot1 + 4\cdot3 = 25$$

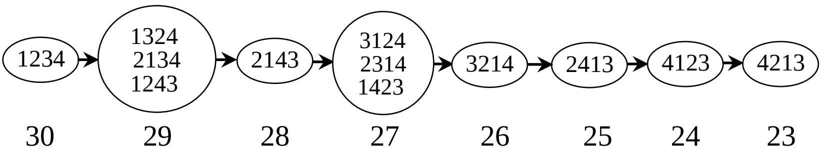
$$F(x_5) = 1\cdot2 + 2\cdot1 + 3\cdot4 + 4\cdot3 = 28$$

$$F(x_{11}) = 1\cdot4 + 2\cdot1 + 3\cdot2 + 4\cdot3 = 24$$

$$F(x_6) = 1\cdot3 + 2\cdot1 + 3\cdot2 + 4\cdot4 = 27$$

$$F(x_{12}) = 1\cdot4 + 2\cdot2 + 3\cdot1 + 4\cdot3 = 23$$

В результаті вище висловлених міркувань, точки на підграфах *A* і *B* можна розташувати в наступний ланцюжок в залежності від значень цільової функції:



Розглянемо точки на підграфі *C*, зокрема $(3\ 1\ 4\ 2)$, $(1\ 4\ 3\ 2)$,

оскільки $\begin{array}{cccc} 3 & 1 & 4 & 2 \\ -1 & 4 & 3 & 2 \\ \hline +2 & -3 & 1 & \end{array} = -2\Delta_1 + \Delta_2 = -\Delta_1 < 0,$ то значення в $(3\ 1\ 4\ 2)$ більше значення в точці $(1\ 4\ 3\ 2)$.

Аналогічно із значеннями в точках $(4\ 1\ 3\ 2)$ і $(3\ 4\ 1\ 2)$,

оскільки $\begin{array}{cccc} 4 & 1 & 3 & 2 \\ -3 & 4 & 1 & 2 \\ \hline +1 & -3 & +2 & \end{array} = -\Delta_1 + 2\Delta_2 = \Delta_2 > 0.$

Виникає співвідношення $-\Delta_1 + 2\Delta_2$, яке необхідне вирішити. Якщо розглядати відношення точок з різних гіперплощин, то значення в точці $(1\ 3\ 4\ 2)$ з B дорівнює значенню в точці $(1\ 4\ 2\ 3)$ з B, оскільки

$$\begin{array}{cccc} 1 & 3 & 4 & 2 \\ -1 & 4 & 2 & 3 \\ \hline & -1 & 2 & -1 \end{array} = \Delta_2 - \Delta_3 = 0\,.$$

Аналогічна ситуація з точками $(3\ 2\ 1\ 4)$ з A і $(1\ 4\ 3\ 2)$ з B, оскільки значення співвідношення дорівнює

$$\begin{array}{cccc} 3 & 2 & 1 & 4 \\ -1 & 4 & 3 & 2 \\ \hline +2 & -2 & -2 & +2 \end{array} = -2\Delta_1 + 2\Delta_2 = 0.$$

Ще розглянемо точки $(2\ 4\ 1\ 3)$ з B і $(1\ 4\ 3\ 2)$ з C і їх співвідношення:

$$\begin{array}{cccc} 2 & 4 & 1 & 3 \\ -1 & 4 & 3 & 2 \\ \hline +1 & -2 & +1 & \end{array} = -\Delta_1 + \Delta_3 = 0,$$

а також точки $(2\ 4\ 1\ 3)$ з B і $(3\ 1\ 4\ 2)$ з C, їх значення рівні,

оскільки $\begin{array}{cccc} 2 & 4 & 1 & 3 \\ -3 & 1 & 4 & 2 \\ \hline -1 & +3 & -3 & 1 \end{array} = \Delta_1 - 2\Delta_2 + \Delta_3 = 0.$

Маємо також рівність значень в точках $(4\ 2\ 1\ 3)$ з B і $(4\ 1\ 3\ 2)$ з C, оскільки

$$\begin{array}{cccc} 4 & 2 & 1 & 3 \\ -4 & 1 & 3 & 2 \\ \hline +1 & -2 & +1 & \end{array} = -\Delta_2 + \Delta_3 = 0.$$

Розглядаючи гіперплощину *D*, знаходимо наступні співвідношення між точками $(2\;3\;4\;1)$ з *D* дорівнює значенню $(4\;1\;2\;3)$ з *B*, оскільки $\begin{array}{cccc} 2 & 3 & 4 & 1 \\ -4 & 1 & 2 & 3 \\ \hline -2 & +2 & +2 & -2 \end{array} = -\Delta_1 + \Delta_3 = 0.$

Аналогічна ситуація з точками $(4\;1\;3\;2)$ з *C* і $(3\;2\;4\;1)$ з *D*, оскільки $\begin{array}{cccc} 4 & 1 & 3 & 2 \\ -3 & 2 & 4 & 1 \\ \hline +1 & -1 & -1 & +1 \end{array} = -\Delta_1 + \Delta_3 = 0,$ а також з точками $(4\;3\;1\;2)$ з *C* і $(3\;4\;2\;1)$ з *D*, оскільки $\begin{array}{cccc} 4 & 3 & 1 & 2 \\ -3 & 4 & 2 & 1 \\ \hline +1 & -1 & -1 & +1 \end{array} = -\Delta_1 + \Delta_3 = 0.$

В результаті розрахунків одержимо гамільтонів шлях графа перестановочного многогранника і впорядкування всіх значень лінійної функції $F(x)$ у порядку їх спадання для точок підграфів *A, B, C, D:*

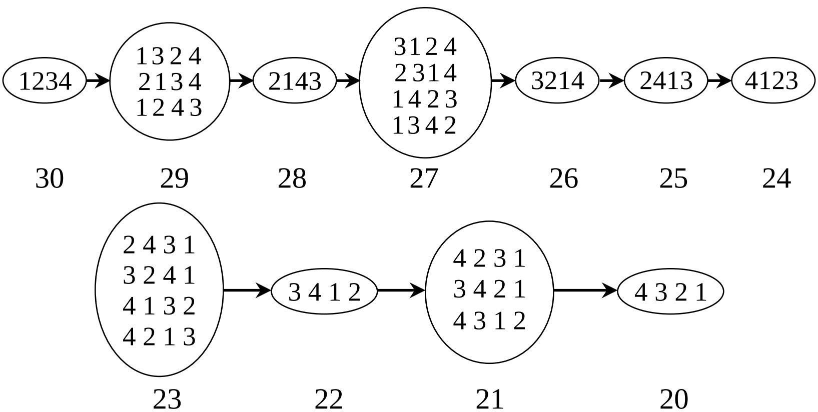


Рис. 4.4. Гамільтонів шлях після знаходження всіх співвідношень лінійної цільової функції $F(x)$

Аналогічно розраховуються і інші варіанти з (10–11).

В процесі визначення гамільтонового шляху виникають три питання, які назвемо α-питаннями. В процесі попередніх обчислень вони вже зустрічалися: $\alpha_1 = \Delta_1 - \Delta_2$ (>0 або <0), $\alpha_2 = \Delta_1 - 2\Delta_2$ (>0 або <0), $\alpha_3 = 2\Delta_1 - \Delta_2$ (>0 або <0). Питання мають деяку залежність, яку можна позначити співвідношеннями:

1) Якщо $\alpha_1 > 0 \rightarrow \alpha_3 > 0$ $\alpha_2 = ?$

2) Якщо $\alpha_1 > 0 \rightarrow \alpha_2 < 0$ $\alpha_3 = ?$

3) Якщо $\alpha_1 = 0 \rightarrow \alpha_2 < 0 \rightarrow \alpha_3 > 0$

В даному випадку $\alpha_1 > 0;$ $\alpha_3 > 0;$ $\alpha_2 = ?$ – невизначений.

Загальна схема на рис. 4.4 перетвориться в схему на рис. 4.5, на якій, якщо $\alpha_i > 0$, то напрями стрілки зберігається.

Тут додані ще шість стрілок, які позначені як β-питання. Ці питання виникають при порівнянні вершин, що належать різним сусіднім підграфам. В даному випадку необхідно з'ясувати тільки одне співвідношення: $\sigma\beta = -\Delta_1 + 2\Delta_2 + \Delta_3$ точні його знак (>0 або <0). При інших розв'язаннях α - питань можуть з'явитися додаткові β-питання.

Для побудови гамільтонового шляху існуючих дуг недостатньо. Необхідно ще порівняти вершини в сусідніх стовпцях, наприклад, пари 2314 і 3142, 2413 і 3241. Питання такого типу називатимемо γ-питаннями.

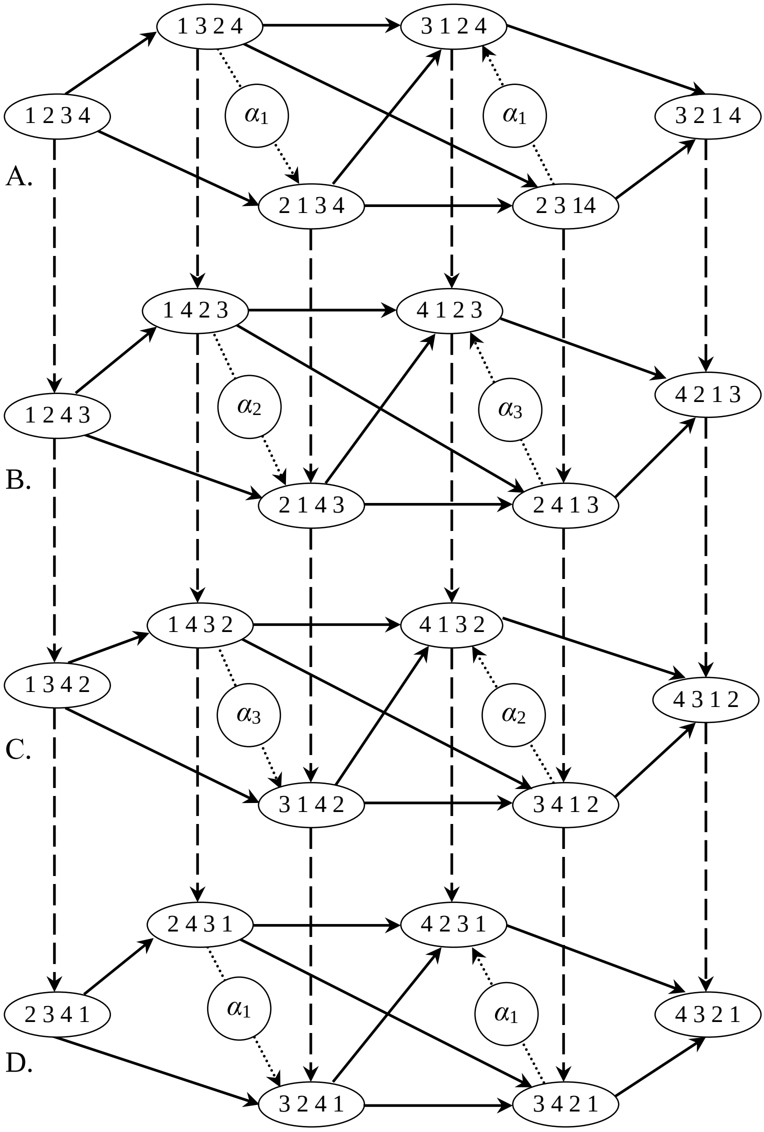


Рис. 4.5. Гамільтонів шлях і α-питання

Якщо $\alpha_2 > 0$, то загальний граф набуває вигляд рис. 4.6.

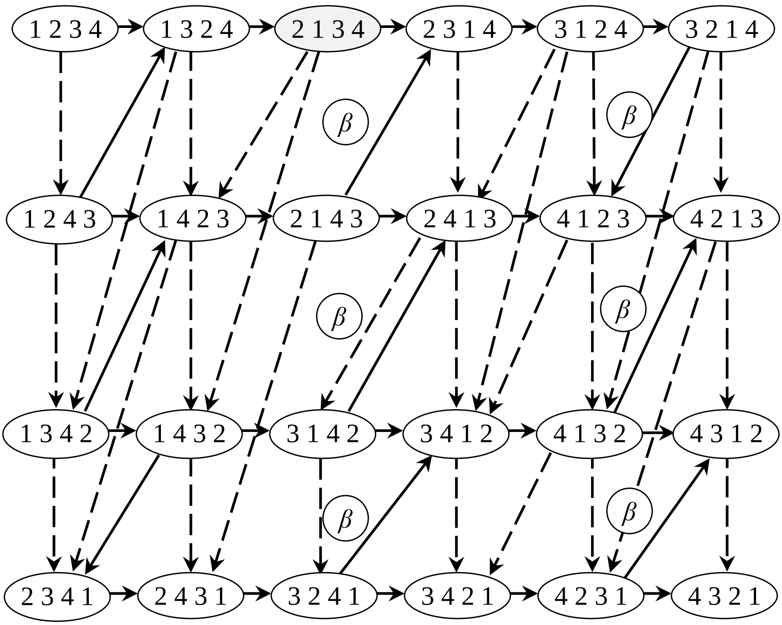


Рис. 4.6. Загальний граф з позначеними β-питаннями

Вище було розглянуто приклад для випадку $n = 4$, але результати можна узагальнити для $n = 5$, а також для довільного значення n. Так в множині перестановок $P(A)$ розглядається 5!=120 точок, які розташовані на 5 гіперплощинах вигляду А (рис. 4.2) і містять по 24 точки кожна. Схематично це можна зобразити на рис. 4.7.

Відповідно, необхідно розгляди α-питань усередині кожного підграфа *A*, *B*, *C*, *D*, *E*, які мають вигляд рис. 4.2.

Для довільної n розмірності матимемо ту ж ситуацію, що і у попередньому випадку, тобто переставний многогранник завжди міститиме підграфи вигляду *A*, тому буде необхідність розглядати α і β-питання.

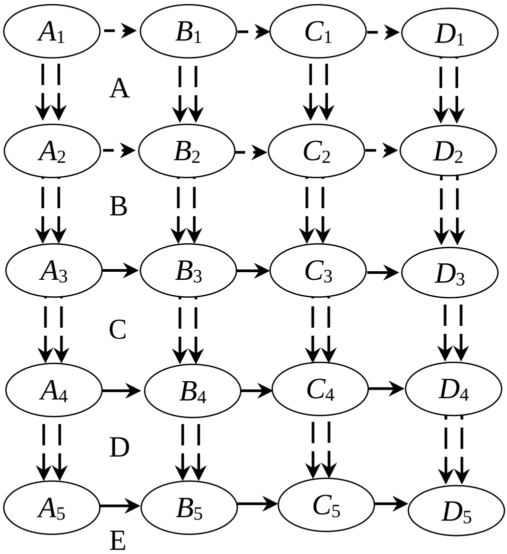


Рис. 4.7. Схематичний граф для $n = 5$

Розглянемо приклад, в якому елементи мультимножини повторюються.

Приклад 4.2. Нехай, дана множина $A = \{1, 2, 2, 4\}$, за допомогою якої утворюється множина перестановок з повтореннями $P_{nk}(A)$, де n – загальна кількість елементів $P_{nk}(A)$, k – різних.

Визначена функція $F(x) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4$, коефіцієнти якої впорядковані таким чином $c_1 \le c_2 \le c_3 \le c_4$ і приймають значення $\{1, 2, 3, 4\}$, а $y^* = F(x^*)$ – значення функції в деякій точці. Необхідно знайти

$$x^* = arg\, extrF(x), \text{ де } x^* \in P(A).$$

Розв'язання. Представимо розкладання графа переставного многогранника $\Pi(A)$ за гіперплощинами. Слід зазначити, що кількість повторень в загальній множині перестановок $P_{nk}(A)$ приводить до «склеювання» гіперплощин, на яких розміщені точки. Визначимо значення цільової функції в кожній вершині многогранника і побудуємо гамільтонів шлях, а також дослідимо взаємозв'язки між гіперплощинами. Тут a_i – відповідні перестановки на рис. 4.4.

$$F(a_1) = 1\cdot1 + 2\cdot2 + 3\cdot2 + 4\cdot4 = 27$$

$$F(a_7) = 1\cdot4 + 2\cdot2 + 3\cdot1 + 4\cdot2 = 19$$

$$F(a_2) = 1\cdot2 + 2\cdot1 + 3\cdot2 + 4\cdot4 = 26$$

$$F(a_8) = 1\cdot2 + 2\cdot2 + 3\cdot1 + 4\cdot4 = 25$$

$$F(a_3) = 1\cdot2 + 2\cdot2 + 3\cdot1 + 4\cdot4 = 25$$

$$F(a_9) = 1\cdot2 + 2\cdot4 + 3\cdot1 + 4\cdot2 = 21$$

$$F(a_4) = 1\cdot1 + 2\cdot2 + 3\cdot4 + 4\cdot2 = 25$$

$$F(a_{10}) = 1\cdot2 + 2\cdot2 + 3\cdot4 + 4\cdot1 = 22$$

$$F(a_5) = 1\cdot1 + 2\cdot4 + 3\cdot2 + 4\cdot2 = 23$$

$$F(a_{11}) = 1\cdot2 + 2\cdot4 + 3\cdot2 + 4\cdot1 = 20$$

$$F(a_6) = 1\cdot4 + 2\cdot1 + 3\cdot2 + 4\cdot2 = 20$$

$$F(a_{12}) = 1\cdot4 + 2\cdot2 + 3\cdot2 + 4\cdot1 = 18$$

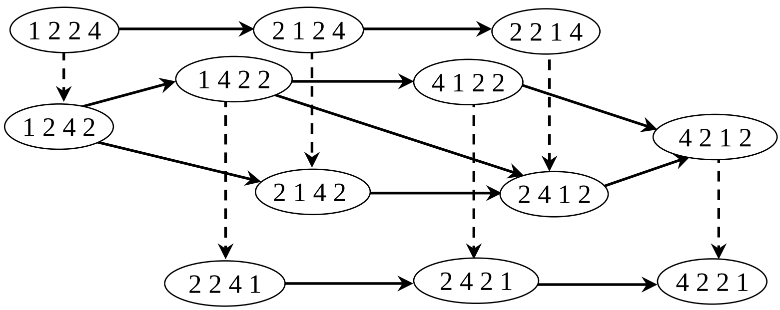


Рис. 4.8. Розкладання точок множини перестановок з повтореннями за значеннями цільової функції

Складемо схему, яка буде відображати розташування точок множини перестановок з повтореннями $P_{nk}(A)$ за значеннями цільової функції на гіперплощинах *A, B, C*, що дасть можливість зобразити гамільтонів шлях для точок всього переставного многогранника $M(A)$. В результаті розрахунків маємо гамільтонів шлях і впорядкування всіх значень лінійної функції $F(x)$:

1 2 2 4 →	2 1 2 4 →	2 2 1 4 →	1 2 4 2 →	2 1 4 2 →	1 4 2 2 →
27	26	25	25	24	23

2 2 4 1 →	2 4 1 2 →	4 1 2 2 →	2 4 2 1 →	4 2 1 2 →	4 2 2 1
22	21	20	19	18	17

Рис. 4.9. Гамільтонів шлях після визначення значень лінійної цільової функції $F(x)$

Переважна кількість задач на множині перестановок є *NP* – повними. Описаний підхід дає можливість використовувати специфічні властивості перестановок, на яких задана лінійна цільова функція. Незалежно від коефіцієнтів функції граф її значень має стандартне часткове упорядкування. Для побудови гамільтонового шляху необхідно до цього графу додати ще декілька дуг, для чого треба зробити відповідні обчислення значень функції. Складність розв’язання задач залежить від кількості цих додаткових обчислень.

4.3. Алгоритм пошуку значень лінійної функції на лексикографічно упорядкованих перестановках

В принципі задачі на комбінаторній множині перестановок можна розв’язувати, якщо множині перестановок $P(A)$ поставити у відповідність повний орієнтований граф, в якому перестановки представлені як вершини, а дві вершини $p_1, p_2 \in P_n(A)$, сполучені дугою, що йде від p_1 до p_2, якщо $F(p_1) \geq F(p_2)$. Очевидно, що максимальне значення функція приймає у вершині, в яку не входить жодна дуга, а мінімальне –

у вершині, з якої не виходить жодна дуга. Гамільтонів шлях в цьому графі відповідатиме послідовності перестановок, для яких значення функції утворює незростаючу послідовність. Якщо перенумерувати цю послідовність, то задачу (4.1)′ можна розв'язати шляхом дихотомії послідовності перестановок до тих пір, поки шуканий аргумент для заданого значення функції не потрапить в інтервал мінімальної довжини. Проте простота алгоритму при його реалізації натрапляє на непереборні труднощі обчислювального характеру, якщо врахувати ту обставину, що число вершин графа, який необхідно побудувати, рівне $n!$. Для визначення напряму дуг графа необхідно виконати $n!$ обчислень значень функції F і C_m^2 порівнянь цих значень, де $m = n!$. Навіть для порівняно невеликих значень n реалізувати цей алгоритм практично нереально. Для подолання цих труднощів побудуємо спеціальний підграф шуканого повного графа. При цьому напрям частини дуг між деякими вершинами можна визначити, не обчислюючи в них значення функції. Замінимо $P(A)$ на $uP(A)$, де u – перестановка, що переводить коефіцієнти $F(x)$ в неспадаючу послідовність. Тому надалі, не порушуючи загальності подальших міркувань, вважатимемо, що $c_1 \le c_2 \le \ldots \le c_u$. Згідно лемі 4.1, якщо з перестановки $p_1 = (i_1, i_2, \ldots, i_n) \in P(A)$ одержана перестановка $p_2 \in P(A)$ транспозицією двох чисел $i_k < i_l$, де $k < l$, то $F(p_1) \ge F(p_2)$. Дійсно

$$F(p_1) - F(p_2) = c_k i_k + c_l i_l - c_k i_k = (i_l - i_k)(c_l - c_k) \ge 0,$$

оскільки обидві величини в дужках невід'ємні.

Побудуємо шуканий граф G_n по індукції. Для $n = 2$ граф містить дві вершини (1,2) і (2,1). По лемі 4.1 від першої вершини до другої виходить дуга. Якщо побудовано граф G_n, то граф G_{n+1} утворюється як $n+1$ копій графа G_n за правилом: у першій копії графа G_n кожній перестановці (вершині) справа приписується число $n+1$. Друга копія виходить з першої транспозицією у всіх перестановках чисел $n+1$ і n. І так далі – з j-й копії транспозицією чисел $(n+2-j)$ і $(n+1-j)$, де $1 \le j \le n$, одержуємо $(j+1)$-у копію. При цьому від всіх вершин j-ї копії йдуть дуги в відповідні вершини $(j+1)$-ї копії. Приклад побудови графів приведено на рис. 4.10. На цьому графі всі дуги побудовані за лемою 4.1.

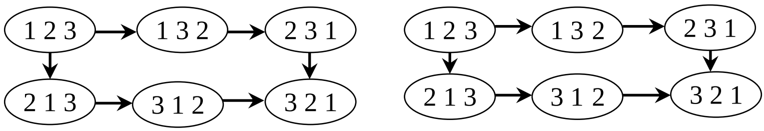


Рис. 4.10

Дуже важливою задачею, яка виникає при побудові залежності між елементами перестановки за значеннями лінійної цільової функції, є задача організації впорядкування і вибору елементу перестановки. Розглянемо алгоритми, які дають можливість встановити взаємозалежність між впорядкуванням елементів перестановки та їх номером.

Алгоритм відображення за номером перестановки N елементу перестановки $P(A)$.

У даному алгоритмі елементи перестановок нумеруються в лексикографічному порядку, який будується за правилом: з двох перестановок

$$a=(a_1,a_2,\ldots,a_k,\ldots,a_n),\ b=(b_1,b_2,\ldots,b_k,\ldots,b_n),\ b>a,$$

якщо знайдеться таке k, при якому $b_k>a_k$, а $b_i=a_{i,}\ i>k$, $(1<k\le n, i\le n)$. Алгоритм дає можливість знайти за заданим номером $1\le N\le n!$ відповідну перестановку. Розглянемо роботу алгоритму на прикладі для $n=4$. Побудуємо послідовність перестановок за вищевикладеним правилом.

Таблиця 4.1

Послідовності перестановок

1)	1234		7)	1243		13)	1342		19)	2341
2)	2134		8)	2143		14)	3142		20)	3241
3)	1324		9)	1423		15)	1432		21)	2431
4)	3124		10)	4123		16)	3412		22)	4231
5)	2314		11)	2413		17)	3412		23)	3421
6)	3214		12)	4213		18)	4312		24)	4321

Задача 4.1. Задано: n – кількість елементів у множині, тоді $n!$ – кількість перестановок, N – номер елементу перестановки.

Знайти: елемент перестановки $x = (x_1, x_2, \ldots, x_n)$.

Початкові дані: $k = n,\ N_0 = N$ масив $r = (1, 2, \ldots, n)$.

Алгоритм працює за кроками: $\Delta = (k-1)!;$ $j = \left|\dfrac{N_0 - 1}{\Delta}\right|;$ $x_k = r_{k-j}$. В масиві r викреслюється елемент r_{k-j}. $N_0 = N_0 - j \cdot \Delta;$ $k = k-1;$ якщо $k = 1$, то $x_1 = r_1$ і кінець. В термінах програмної реалізації алгоритм для $N = 19$ виглядатиме таким чином: $k = 4; r = [1234]$.

$i = 1; \Delta = 3! = 6;$	$i = 2, \Delta = 2;$	$i = 3, \Delta = 1;$
$j = \left\lvert\dfrac{19-1}{6}\right\rvert = 3;$	$j = \left\lfloor\dfrac{1-1}{2}\right\rfloor = 0,$	$j = \left\lfloor\dfrac{1-1}{1}\right\rfloor = 0,$
$x_4 = r_{4-3} = 1;$	$x_3 = r_{3-1} = 4;$	$x_2 = r_{2-0} = 3;$
$r = [234];$	$r = [23];$	$r = [2];$
$N_0 = 19 - 3 \cdot 6 = 1;$	$N_0 = 1 - 2 \cdot 0 = 1;$	$N_0 = 1 - 1 \cdot 0 = 1;$
$k = 3;\ i = 2.$	$k = 2, i = 3$	$k = 1; i = 4; x_1 = 2.$

В результаті одержимо перестановку $x = (2, 3, 4, 1)$, що відповідає номеру елемента в таблиці.

Алгоритм відображення номера перестановки *N* за елементом перестановки $\boldsymbol{P(A)}$. Даний алгоритм дає можливість розв'язати обернену задачу: за заданою перестановкою знайти її номер в лексикографічно впорядкованій послідовності перестановок. Розглянемо реалізацію даного алгоритму на прикладі.

Нехай задана перестановка $x = \left(x_{i_1}, x_{i_2}, \ldots, x_{i_n}\right)$. Знайти: N – номер елементу перестановки. Початкові дані: $k = n, N = 0,$ масив. $r = (1, 2, \ldots, n)$. Робота алгоритму: за значенням x_k знаходимо номер елементу j в масиві r і викреслюємо його. Обчислюємо $N = (k-1)!(k-j) + N; k = k-1;$ якщо $k = 1,$ то $N = N + 1$ і кінець.

В термінах програмної реалізації алгоритм виглядатиме таким чином при початкових параметрах: $n = 4$, $x_n = (2, 4, 3, 1)$, $N = 0$.

$$r = (1,2,3,4), k = n = 4.$$

$$k = 4, x_4 = 1, j = 1, \ N = 3!(4-1) = 18; \ r = [234];$$

$$k = 3, x_3 = 3, \ j = 2, \ N = 18 + 1 \cdot 2! = 20; \ r = [24];$$

$$k = 2, x_2 = 4, j = 2, \ N = 20 + 0 = 20; \ r = [2 \ ];$$

$$k = 1, \ x_1 = 2, \ j = 4, \ N = 20 + 1 = 21.$$

В результаті одержимо $N = 21$.

4.4. Горизонтальний метод локалізації значення лінійної функції, заданої на перестановках

Далі розглянемо граф $G(P_n)$ переставного многогранника, в якому вершини є множиною всіх перестановок P_n, а дві вершини утворять дугу $\overrightarrow{p_1 p_2}$, якщо $f(p_1) \geq f(p_2)$, $p_1, p_2 \in P_n$ і якщо перестановка p_2 одержана з p_1 за допомогою транспозиції двох елементів.

В даному підрозділі розглядається задача (4.2) − локалізація значень лінійної функції на перестановках – та пропонується метод її розв’язання.

Є множина, що визначає граф перестановок, ставиться задача відшукання підмножини перестановок, на яких досягається. задане значення функції.

Якщо не існує жодної такої перестановки, то тоді необхідно розв’язувати задачу (4.2), що визначена для деякої підмножини перестановок. Якщо сформулювати задачу локалізація значення функції в термінах теорії графів, то вона буде звучати наступним чином: для деякої перестановки p_i відомо значення лінійної цільової функції; тоді для задачі (4.2) необхідно визначити відповідну вершину, чи множину вершин графа $G(P_n)$ перестановок, в яких це значення досягається; а для задачі (4.2) необхідно визначити множину ребер $\{(p,q)\}$ цього графа таких, що $f(q) < f(p_i) < f(p)$.

При розв'язуванні даної задачі важливим є визначення підмножини перестановок – послідовності підграфів $G_1, G_2, \ldots, G_s$, на яких буде розглядатися пошук необхідного значення. Після попадання на множину вершин деякого підграфа G_1 переходимо до підграфа G_2 з довільної вершини $p_i \in P_n$, якщо є необхідним визначення заданого значення на ньому. Як видно з наведених рисунків графів перестановок, всі вони, незалежно від величини n мають ієрархічний вигляд, де підграфи з більшим значенням функції у вершинах знаходяться вище, ніж підграфи з меншим значенням функції у вершинах. Кожні такі підграфи, в чвою чергу, містять зліва максимальне значення функції, а справа – мінімальне значення.

Це наводить на думку про те, що при розв'язуванні задачі локалізації функції необхідно спочатку розглядати лише ці крайні значення підграфів, а при необхідності – проміжкові значення між ними. Звідси і зрозуміло, чому даний метод локалізації значень функції дістав назву горизонтального.

Визначення 4.1. Назвемо підграфом *r*-ранга графа переставного многогранника $G(P_u)$ граф, вершини якого мають *r* фіксованих старших координат.

З цієї точки зору граф $G(P_u)$ є підграфом 0-ранга многогранника $G(P_u)$ і складається з *n* підграфів 1-ранга $G(P_{n-1})$, де старша координата дорівнює відповідно $n, n-1, n-2, \ldots, 2, 1$. Аналогічно можна класифікувати і підграфи більших рангів.

Відомо, що максимальне значення лінійна функція $f(x)$ на переставному многограннику $G(P_n)$ приймає в перестановці $(1, 2, \ldots, n)$, а мінімальне – в перестановці $(n, n-1, \ldots, 2, 1)$, якщо коефіцієнти цільової функції впорядковані за збільшенням. Очевидно, що ця властивість зберігається і для підграфів, утворених підмножиною перестановок, у яких фіксовані останні *k* елементів.

Очевидно, що обидві задачі локалізації – (4.1) – можна вирішити, якщо на графі $G(P_n)$ добудувати множину дуг між несуміжними перестановками, які дозволять пройти по дугам шлях від початкової вершини, де $f(x)$ приймає максимальне значення, до кінцевої, де $f(x)$ приймає мінімальне значення, тобто побудувати між ними гамільтонів шлях, як це описано в попередньому підрозділі та в роботі [60]. Якщо відома послідовність перестановок, через які він проходить, то за допомогою методу дихотомії гамільтонового шляху, обчислюючи значення функції у відповідній перестановці, завжди можна локалізувати довільне значення цільової функції $f(x)$. З погляду теорії графів, екстремальну комбінаторну задачу оптимізації можна сформулювати наступним чином: серед деякої множини шляхів *L,* що з'єднують дві задані вершини, відшукати мінімальний (або максимальний), тобто шлях, що має мінімальне (або максимальне) значення λ. Як множина *L* може бути вибрана, наприклад, множина всіх гамільтонових шляхів. Розглянемо теорему, яка дає можливість визначення мінімального і максимального шляху серед всіх простих шляхів, що сполучають дві фіксовані вершини.

Теорема 4.1. [122, 203] Нехай деякий шлях, що сполучає вершину x рівня m і вершину x' рівня s, є мінімальним (максимальним). Тоді його підшлях між вершиною y рівня k і вершиною y' рівня p $(m \le k < p \le s)$ також є мінімальним (максимальним).

Наведена теорема лежить в основі методу відшукання максимальних шляхів в графі без контурів і дає можливість при побудові алгоритму локалізації значень розглянути підграфи графа $G(P_n)$. Задачі (4.1) та (4.2) розглядаються на вершинах графа, тоді знаходження відповідної вершини розглядаємо послідовно, починаючи з деякої початкової, а потім всі вершини графа у порядку спадання значень цільової функції в цих вершинах і приписуємо кожній вершині число, рівне деякому значенню функції $f(x)$. Для відшукання мінімальних шляхів у графах, що мають контури, також існують різні методи. Розглянемо далі підхід до розв'язання задач (4.1) та (4.2), що дозволяє вибирати лише частину вершин графа і знаходити необхідне значення.

Визначення 4.2. Назвемо схему зображення графа перестановок $G(P)_n$, в якій кожен з n підграфів 1-рангa $G(P_{n-1})$ зображений у вигляді одного ребра, що з'єднує дві вершини з максимальним та мінімальним значеннями функції, ***структурним графом*** перестановок.

Це поняття має об'єктивний зміст, і його використання лягло в основу алгоритму реалізації горизонтального методу локалізації значення лінійної функції цілі.

Алгоритм локалізації значення лінійної функції на перестановках. Початковий крок:

1) вводимо n – кількість елементів перестановки і розмірність цільової функції;

2) задаємо значення коефіцієнтів $c_1, c_2, \ldots, c_u$ цільової функції $f(x)$, нормалізуючи цільову функцію при потребі;

3) вводимо значення елементів множини перестановок: $a_1, a_2, \ldots, a_n$ причому введення здійснюється так, щоб елементи були впорядковані по наступному співвідношенню: $a_1 \le a_2 \le \ldots \le a_n$;

4) задаємо значення цільової функції $y_0 = f(x)$.

Крок 1. Обчислюємо $n!$.

Крок 2. Визначаємо значення: $b = (n-1)!$,, яке характеризує кількість точок в загальному структурному графі в кожному підграфі.

Під загальним структурним графом розуміємо граф, що відображає часткову впорядкованість елементів конфігурації за значеннями цільової функції і містить лише вершини, що розміщені на еквівалентних підграфах і в яких досягаються лише екстремальні значення функції.

Крок 3. Обчислюємо мінімальне і максимальне значення заданої функції $f(x)$: $f(x)_{max} = c_1x_1 + c_2x_2 + \ldots + c_{n-1}x_{n-1} + c_nx_n$;

$$f(x)_{min} = c_nx_1 + c_{n-1}x_2 + \ldots + c_2x_{n-1} + c_1x_n.$$

Крок 4. Будуємо структурний граф, який має n підграфів, а на кожному підграфі – початкову і кінцеву вершини.

Крок 5. Визначаємо значення цільової функції $f(x)$ у точках – вершинах підграфів $x = \left(x_{i_1}, x_{i_2}, \dots, x_{i_n}\right)$.

Крок 6. Визначаємо зліва в структурному графі множину точок, для яких виконується умова $f\left(\overline{x}_i\right) \geq f\left(x^0\right)$, де x^0 точка, для якої $f\left(x^0\right) \geq y^0$, а $\overline{x}_i$, $i \in N_k$, множина точок, значення цільової функції $f(x)$ у яких більше заданого.

Крок 7. Справа визначимо множину точок, які дають значення цільової функції згідно з виконанням умови: $f\left(\overline{x}_j\right) \leq f\left(x^0\right)$, де x^0 – точка, для якої виконується умова $f\left(x^0\right) \leq y^0$; а $\overline{x}_j$, $j \in N_{n-k}$, множина точок, для яких значення цільової функції $f(x)$ менше заданого.

Крок 8. Визначаємо множину підграфів як множину перетинів, елементи якого задовольняють умовам, згідно кроку 6 та 7.

Крок 9. Формуємо множину точок − елементів перестановки, що задовольняють умову $f\left(x_0\right) = y_0$. Якщо всі точки знайдені, тобто серед множини підграфів, визначених на кроці 8, немає таких, для вершин яких виконувалася б умова $f\left(\overline{x}_j\right) \leq y_0 \leq f\left(\overline{x}_j\right)$, то задача розв'язана. Здійснюється вибір елементів точок перестановки. Інакше перехід на наступний крок.

Крок 10. Визначаємо підграф, для якого виконується умова $f\left(\overline{x}_j\right) \leq y_0 \leq f\left(\overline{x}_j\right)$. Фіксуємо останню координату в точці, вершині підграфа.

Крок 11. Покладаємо $n := n-1$. Здійснюємо перехід на крок 1.

Слід зазначити, що генерація точок – вершин перестановок в крайніх вершинах підграфів здійснюється рекурсивним методом, що описаний в другому розділі, причому з $n!$ елементів необхідно згенерувати тільки $2n$ на початковому етапі.

Алгоритм був програмно реалізований, проведені чисельні експерименти. Нижче наведено чисельний приклад описаного алгоритму:

Приклад 4.4. Дано: а) функція $f(x)=c_1x_1+c_2x_2+...+c_nx_n$, де $n=6$, тоді $f(x)=4x_1+8x_2+2x_3+7x_4+3x_5+6x_6$;

б) визначене значення функції $f(x^*)=109$;

в) елементи множини перестановки $(1\ 2\ 3\ 4\ 5\ 6)$.

Знайти: точки – вершини переставного многогранника $x^*=?$, в яких досягається задане значення цільової функції.

Розв'язання: Знаходимо перетворення для впорядкування коефіцієнтів цільової функції, якщо вони були не впорядковані.

$$f(x)=\tilde{c}_1x_1+...+\tilde{c}_nx_n,\ \tilde{c}_1\le\tilde{c}_2\le\ldots\le\tilde{c}_n,$$

$$c=(4,\ 8,\ 2,\ 7,\ 3,\ 6),\ \tilde{c}=(2,3,4,6,7,8),$$

$$\tilde{c}_i=c_{\pi^{-1}},\ \pi=\begin{pmatrix}1&2&3&4&5&6\\3&6&1&5&2&4\end{pmatrix},\ \pi^{-1}=\begin{pmatrix}1\,2\,3\,4\,5\,6\\3\,5\,1\,6\,4\,2\end{pmatrix}:$$

$$\tilde{c}_1=c_3;\tilde{c}_2=c_5;\tilde{c}_3=c_1;\tilde{c}_4=c_6;\tilde{c}_5=c_4;\tilde{c}_6=c_2.$$

Будуємо структурний граф для $n=6$. Всього вершин у графі $G(P_6)=6!=720$, а у структурному графі – 12. Визначимо максимальне і мінімальне значення лінійної функції на множині перестановок

$$max:=2\cdot1+3\cdot2+4\cdot3+6\cdot4+7\cdot5+8\cdot6=127$$

$$min:=2\cdot6+3\cdot5+4\cdot4+6\cdot3+7\cdot2+8\cdot1=83.$$

Аналогічно визначимо мінімум і максимум в решті крайніх вершин кожного підграфа і зобразимо це на рисунку.

На рис. 4.11 зображений структурний граф, де відмічені крайні точки гіперплощин (підграфів), Зліва знаходяться вершини, в яких досягається максимальні значення цільової функції на підграфі, справа – вершини, в яких досягається мінімальне значення функцій.

Слід зазначити, що значення функції визначаються підстановкою координат вершин в цільову функцію вигляду

$$f(x)=2x_1+3x_2+4x_3+6x_4+7x_5+8x_6.$$

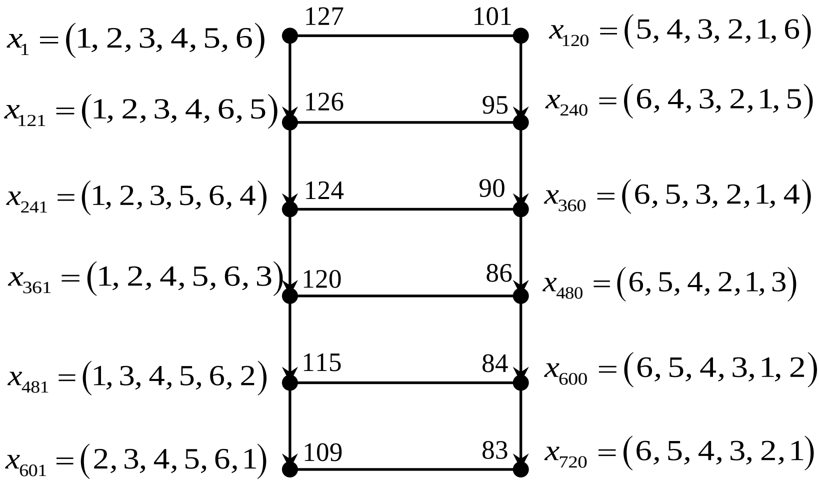


Рис. 4.11. Структурний граф переставного многогранника

Вибір відповідної структури редставлення графів має принциповий вплив на ефективність алгоритмів, тому докладніше зупинимося на цій проблемі. Згідно рис. 4.11 є вершина, для якої шукане значення функції досягається $f(x) = 109$ – це точка $x_{601} = (2,3,4,5,6,1)$ – вершина в нижньому підграфі. Відповідно, в ньому більше не буде вершин – перестановок, в яких досягається таке значення. Значить, пошук наступного значення здійснюється на наступному, що вище стоїть, підграфі, де $84 \le f(x) \le 115$. Для цього підграфа характерною є умова, що остання координата вершин перестановок підграфа рівна $x_6 = 2$. Далі розглянемо цей підграф, зафіксувавши $x_6 = 2$, структурний граф цього підграфа для $n = 5$ і п’ять його підграфів рангу $r = 1$.

Наступний крок: у рис. 4.12 відкидаємо 2 нижні підграфи і беремо три верхні підграфи рангу $r = 2$, далі розглядаємо підграф з вершинами, в яких останні дві координати рівні: $x_6 = 2;$ $x_5 = 4$ оскільки в двох відкинутих не знайдуться вершини, які задовольняють умову $f(x) = 109$.

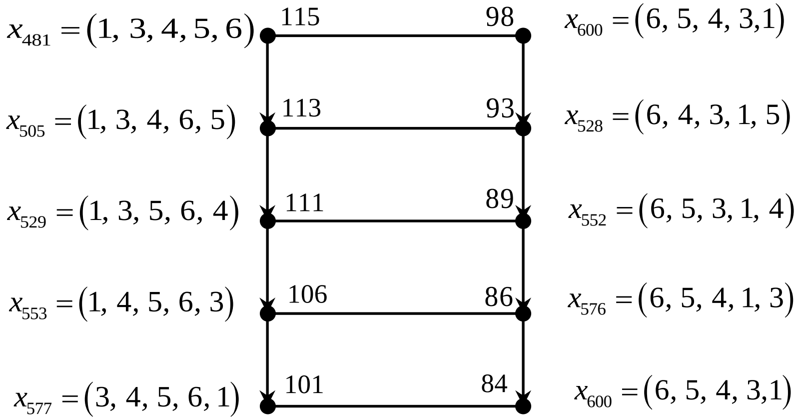


Рис. 4.12. Структурний граф при $n = 5$

Визначимо значення функції $f(x)$ у крайніх точках графа:

$$f(1,3,5,6,4,2) = 2\cdot 1 + 3\cdot 3 + 4\cdot 5 + 6\cdot 6 + 7\cdot 4 + 8\cdot 2 = 111$$

$$f(1,3,6,5,4,2) = 2\cdot 1 + 3\cdot 3 + 4\cdot 6 + 6\cdot 5 + 7\cdot 4 + 8\cdot 2 = 109$$

$$f(1,5,6,3,4,2) = 2\cdot 1 + 3\cdot 5 + 4\cdot 6 + 6\cdot 3 + 7\cdot 4 + 8\cdot 2 = 103$$

$$f(3,5,6,1,4,2) = 2\cdot 1 + 3\cdot 5 + 4\cdot 6 + 6\cdot 1 + 7\cdot 4 + 8\cdot 2 = 95$$

$$f(5,3,1,6,4,2) = 2\cdot 5 + 3\cdot 3 + 4\cdot 1 + 6\cdot 6 + 7\cdot 4 + 8\cdot 2 = 103$$

$$f(6,3,1,5,4,2) = 2\cdot 6 + 3\cdot 3 + 4\cdot 1 + 6\cdot 5 + 7\cdot 4 + 8\cdot 2 = 99$$

$$f(6,5,1,3,4,2) = 2\cdot 6 + 3\cdot 5 + 4\cdot 1 + 6\cdot 3 + 7\cdot 4 + 8\cdot 2 = 93$$

$$f(6,5,3,1,4,2) = 2\cdot 6 + 3\cdot 5 + 4\cdot 3 + 6\cdot 1 + 7\cdot 4 + 8\cdot 2 = 89$$

Тоді на рисунку представимо структурний підграф рангу $r = 2$ при $x_5 = 4;$ $x_6 = 2$ у вигляді:

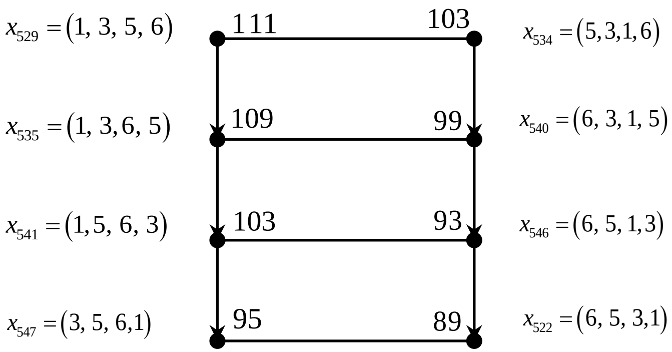


Рис. 4.13. Структурний граф при $n = 4$

Далі, робимо аналогічну процедуру порівняння знайдених значень цільової функції в крайніх вершинах із заданим значенням $f(x) = 109$ згідно умові задачі.

Знову, визначаємо значення цільової функції $f(x)$ у крайніх вершинах графа:

$$f(1, 3, 5, 6, 4, 2) = 2 \cdot 1 + 3 \cdot 3 + 4 \cdot 5 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 111$$

$$f(1, 5, 3, 6, 4, 2) = 2 \cdot 1 + 3 \cdot 5 + 4 \cdot 3 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 109$$

$$f(3, 5, 1, 6, 4, 2) = 2 \cdot 3 + 3 \cdot 5 + 4 \cdot 1 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 105$$

$$f(3, 1, 5, 6, 4, 2) = 2 \cdot 3 + 3 \cdot 1 + 4 \cdot 5 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 109$$

$$f(5, 1, 3, 6, 4, 2) = 2 \cdot 5 + 3 \cdot 1 + 4 \cdot 3 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 105$$

$$f(5, 3, 1, 6, 4, 2) = 2 \cdot 5 + 3 \cdot 3 + 4 \cdot 1 + 6 \cdot 6 + 7 \cdot 4 + 8 \cdot 2 = 103$$

Розглянемо підграф, для вершин якого: $x_4 = 6$, $x_5 = 4$; $x_6 = 2$. На підставі обчислень, одержуємо наступне розташування вершин:

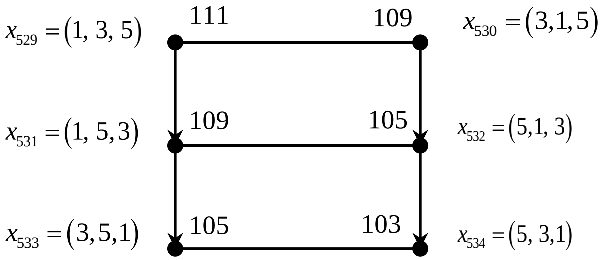


Рис. 4.14. Структурний граф при $n = 3$

На підставі елементарних операцій порівняння одержимо розв’язки для $x_{531}^{*} = (1,5,3,6,4,2)$, $x_{530}^{*} = (3,1,5,6,4,2)$, які досягаються в вершинах структурного підграфа переставного многогранника, в яких досягається значення функції $f(x) = 109$.

Це був розглянутий один нижній підграф при фіксованій координаті $x_6 = 2$ із структурного графа, зображеного на рис. 4.11. Розглянемо третій підграф знизу з рис. 4.11 при $x_6 = 3$.

Проробляємо ту ж процедуру: переглянемо значення цільової функції в крайніх вершинах. Далі розглядається структурний граф, у вершинах якого розташовані точки $[x_{361}; x_{480}]$, при останній координаті $x_6 = 3$ і значення цільової функції в цих вершинах знаходиться в межах $120 \geq f(x) \geq 86$. На підставі обчислень з графа виділяємо підграф, для яких виконується умова $x_5 = 4,\ x_6 = 3$. Графічне представлення наступне:

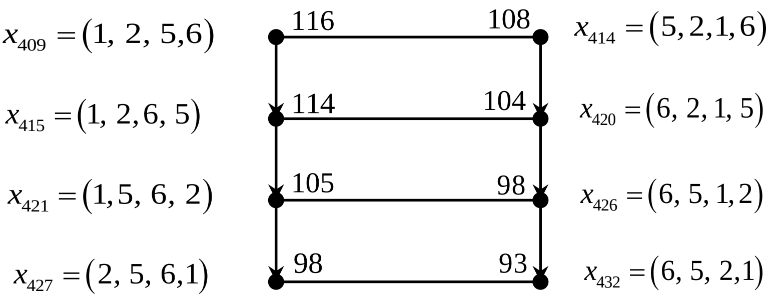


Рис. 4.15. Структурний граф при $n = 4$

На підставі аналізу рис. 4.15 є доцільним розглянути підграф, для вершин $[x_{415};x_{420}]$, значення функції в яких знаходяться в межах $114 \ge F(x) \ge 104$. Далі розглядається розташування вершин на підграфі, в якому зафіксована третя координата, за наявності зафіксованих двох останніх: $x_4 = 5$; $x_5 = 4$; $x_6 = 3$.

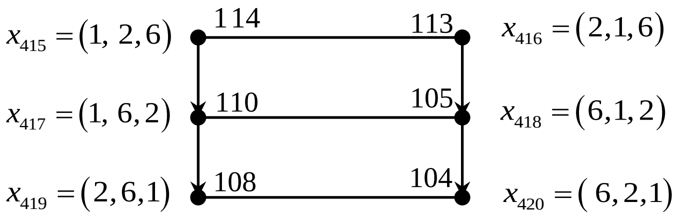


Рис. 4.16. Структурний граф при $n = 3$

Як показали розрахунки, на цьому підграфі немає вершин, що задовольняють значення функції $f(x) = 109$. Тому повертаємося до рис. 4.11 і розглядаємо четвертий знизу підграф, для вершин якого $x_6 = 4$. Визначимо значення цільової функції в крайніх вершинах.

$$f(x_{241}) = 2\cdot 1 + 3\cdot 2 + 4\cdot 3 + 6\cdot 5 + 7\cdot 6 + 8\cdot 4 = 124$$

$$f(x_{264}) = 2\cdot 5 + 3\cdot 3 + 4\cdot 2 + 6\cdot 1 + 7\cdot 6 + 8\cdot 4 = 107$$

$$f(x_{265}) = 2\cdot 1 + 3\cdot 2 + 4\cdot 3 + 6\cdot 6 + 7\cdot 5 + 8\cdot 4 = 123$$

$$f(x_{288}) = 2\cdot 6 + 3\cdot 3 + 4\cdot 2 + 6\cdot 1 + 7\cdot 5 + 8\cdot 4 = 102$$

$$f(x_{289}) = 2\cdot 1 + 3\cdot 2 + 4\cdot 5 + 6\cdot 6 + 7\cdot 3 + 8\cdot 4 = 117$$

$$f(x_{312}) = 2\cdot 6 + 3\cdot 5 + 4\cdot 2 + 6\cdot 1 + 7\cdot 3 + 8\cdot 4 = 94$$

$$f(x_{313}) = 2\cdot 1 + 3\cdot 3 + 4\cdot 5 + 6\cdot 6 + 7\cdot 2 + 8\cdot 4 = 113$$

$$f(x_{336}) = 2\cdot 6 + 3\cdot 5 + 4\cdot 3 + 6\cdot 1 + 7\cdot 2 + 8\cdot 4 = 91$$

$$f(x_{337}) = 2\cdot 2 + 3\cdot 3 + 4\cdot 5 + 6\cdot 6 + 7\cdot 1 + 8\cdot 4 = 108$$

$$f\left(x_{360}\right)=2\cdot 6+3\cdot 5+4\cdot 3+6\cdot 2+7\cdot 1+8\cdot 4=90$$

Позначимо крайні вершини на рис. 4.17:

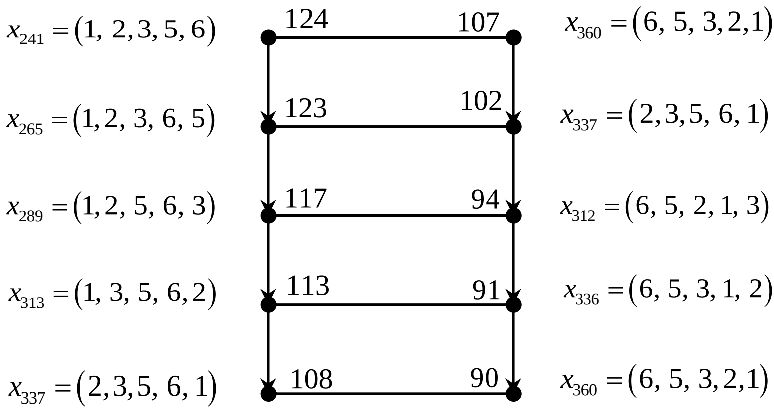


Рис. 4.17 Структурний граф при $x_6 = 4$

На підставі вище зроблених розрахунків для знаходження точки – перестановки доцільно розглянути другий нижній підграф і знову маємо розкладання: при $n = 4$. Визначимо значення цільової функції в крайніх вершинах підграфів і позначимо на рис. 4.18:

$$f\left(x_{313}\right)=2\cdot 1+3\cdot 3+4\cdot 5+6\cdot 6+7\cdot 2+8\cdot 4=113$$

$$f\left(x_{319}\right)=2\cdot 1+3\cdot 3+4\cdot 6+6\cdot 5+7\cdot 2+8\cdot 4=111$$

$$f\left(x_{325}\right)=2\cdot 1+3\cdot 5+4\cdot 6+6\cdot 3+7\cdot 2+8\cdot 4=105$$

$$f\left(x_{331}\right)=2\cdot 3+3\cdot 5+4\cdot 6+6\cdot 1+7\cdot 2+8\cdot 4=97$$

$$f\left(x_{318}\right)=2\cdot 5+3\cdot 3+4\cdot 1+6\cdot 6+7\cdot 2+8\cdot 4=105$$

$$f\left(x_{324}\right)=2\cdot 6+3\cdot 3+4\cdot 1+6\cdot 5+7\cdot 2+8\cdot 4=101$$

$$f\left(x_{330}\right)=2\cdot 6+3\cdot 5+4\cdot 1+6\cdot 3+7\cdot 2+8\cdot 4=95$$

$$f\left(x_{336}\right)=2\cdot 6+3\cdot 5+4\cdot 3+6\cdot 1+7\cdot 2+8\cdot 4=91.$$

На рис. 4.18 представлено схематично структурний граф, де зліва позначені вершини підграфів, в яких досягається максимальне значення функції на кожному з них, справа – вершини, в яких досягається мінімальне значення функції на кожному з підграфів.

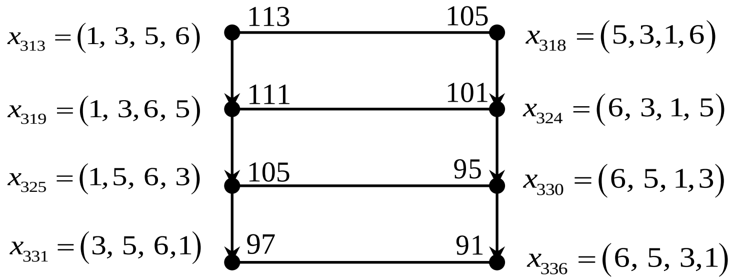


Рис. 4.18. Структурний граф при $x_5=2,\ x_6=4$

Згідно розрахункам, є доцільно розглянути два верхні підграфи. Далі розглядається один з них при фіксованих координатах $x_4=5, x_5=2, x_6=4$ і виконуються вже описані процедури.

На цьому кроці, нарешті, одержуємо розв'язок задачі (4.2), який задовольняє умову $f(x)=109$. Всього отримано вісім вершин, які перелічені нижче. Отже, необхідності розв'язувати задачу (4.1″) не виникло.

$$x_1^0=(2,3,4,5,6,1),\ \ x_2^0=(1,3,6,5,4,2),$$

$$x_3^0=(1,5,3,6,4,2),\ \ x_4^0=(3,1,5,6,4,2),$$

$$x_5^0=(3,1,6,5,2,4),\ \ x_6^0=(3,4,2,6,1,5),$$

$$x_7^0=(4,2,3,6,1,5),\ \ x_8^0=(2,4,5,3,1,6).$$

Цей розв'язок отримано для нових змінних, у яких цільова функція упорядкована. Для того, щоб вернутися до початкових змінних, треба домножити цей розв'язок на перестановку $\pi^{-1} = \begin{pmatrix} 1\,2\,3\,4\,5\,6 \\ 3\,5\,1\,6\,4\,2 \end{pmatrix}$. Після цього отримаємо розв'язок

$$x_1 = (4,1,2,6,3,5);\ x_2 = (6,2,1,4,3,5);\ x_3 = (3,2,1,4,5,6);$$

$$x_4 = (5,2,3,4,1,6);\ x_5 = (6,4,3,2,1,5);\ x_6 = (2,5,3,1,4,6);$$

$$x_7 = (3,5,4,1,2,6);\ x_8 = (5,6,2,1,4,3).$$

Для знаходження розв'язку цієї задачі знадобилося зробити обчислення значень функції в 74 вершинах графа перестановок. Нагадаємо, що всього цих вершин 720. Розглянемо ще один метод розв'язання даної задачі.

4.5. Координатний метод локалізації значення лінійної функції, заданої на перестановках

Слід зазначити, що значення функції на графі перестановок в напрямку знизу вверх зростає, а зверху вниз – спадає з однаковим інтервалом при рівномірному розподілі значень коефіцієнтів. Це обумовлено ієрархічною будовою цього графа та лінійністю цільової функції. При цьому поняття «верх», або «низ» мають стабільний структурний зміст у відношенні підграфів у рамках об'єднуючого графа. Так, наприклад, підграф $G(P_3)$, зображений на рис. 4.1, є складовою частиною всіх графів з більшою кількістю координат. Чотири підграфа $G(P_3)$ складають граф $G(P_4)$, п'ять підграфів $G(P_4)$, або 20 підграфів $G(P_3)$, складають $G(P_5)$ і так далі. Це означає, що для довільного n значення цільової функції в шести вершинах підграфа $G(P_3)$, у яких до координат справа дописані числа $4,5,6,\dots,n-1,n,$ складають шість найбільших серед $n!$ значень функції. Починаючи з однієї з цих вершин, можна визначити множину шляхів, на яких значення функції спадає. Це і є підґрунтям для координатного методу локалізації функції. Розглянемо його суть на прикладі, описаному в п. 4.4. Отже маємо функцію $f\left(x\right)=4x_1+8x_2+2x_3+7x_4+3x_5+6x_6$. Треба знайти перестановки x^*, в яких $f\left(x^*\right)=y^*=109$. Після нормалізації ця функція прийме вигляд:

$$f\left(x\right)=2x_1+3x_2+4x_3+6x_4+7x_5+8x_6.$$

Будемо поступово розглядати підграфи G_i, тобто підграфи, у яких фіксована остання координата – у нашому прикладі $n=6$, а $x_6=1,2,3,4,5,6$, або $x_6\in N_6$. Кожній вершині графа на рис. 4.1, в залежності від її типу, можна поставити у відповідність свій підграф G_i, зображений на рис. 4.19. Під типом вершини розуміється наступне: якщо шоста координата для всього графа постійна (на рис. 4.19 це 2), то п'ята і четверта змінюються від більшої можливої до меншої. Якщо три старші координати вже вибрані, то порядок перших трьох, який відповідає одній з шести вершин на рис. 4.1, і визначає тип вершини. Пояснимо побудову і структуру рис. 4.19.

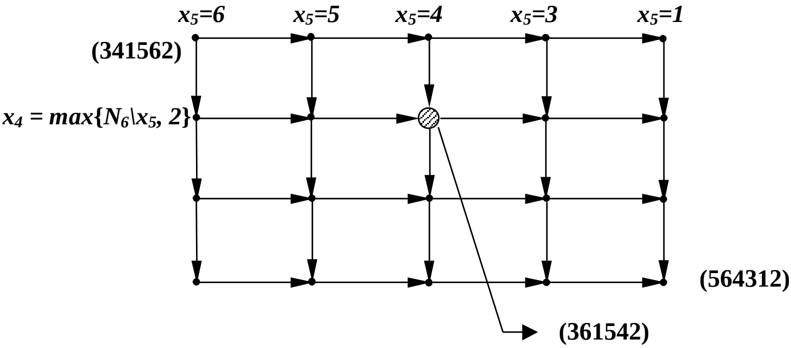


Рис. 4.19. Схема підграфа G_2 для типу вершин (231)

Граф представляє мережу, де витоком є найвища і найлівіша вершина, а стоком – найнижча і найправіша вершина.

Нехай на рис. 4.1 як тип ми вибрали вершину (231). Тоді координати витока такі: $x_6=2, x_5=6, x_4=5$. Упорядкуємо інші координати, що залишилися $i_1=1$, $i_2=3$, $i_3=4$. Тип вершини зобов'язує, щоб нижні індекси перших трьох координат були відповідно (231), тобто код витоку має вигляд (341562). Аналогічно знаходимо координати стоку. Ясно, що тут четверта і п'ята координати вибираються як найменші можливі, тобто $x_6 = 2$, $x_5 = 1$, $x_4 = 3$. Для перших трьох координат маємо $i_1 = 4$, $i_2 = 5$, $i_3 = 6$, і код стоку має вигляд (564312). Для довільної вершини цього підграфа (на рисунку вона. заштрихована) застосовується той же принцип. Для неї $x_6 = 2$, $x_5 = 4$, а $x_4 = 5$ як друге за величиною значення серед залишених 1, 3, 5, 6. Звідси код виділеної вершини (361542).

Тепер будемо розв'язувати поставлену задачу, при цьому дотримуючись таких правил: а) якщо в даній вершині значення функції менше заданого, то далі треба шукати це значення в сусідній вершині, збільшуючи четверту, або п'яту координату; б) якщо в даній вершині значення функції більше заданого, то далі треба шукати це значення в сусідній вершині, зменшуючи четверту, або п'яту координату.

Покажемо, що для розв'язання задачі не обов'язково розглядати всі коди підграфів типу на рис. 4.19 і обчислювати в них значення функції, а достатньо обчислити її значення в витоках мереж, які відповідають підграфам з фіксованим значенням шостої координати. Для прикладу виберемо тип вершини (213). Розглянемо підграфи G_i і для кожного обчислимо значення функції у витоках цих підграфів. Маємо таку таблицю:

Таблиця 4.2

Підграфи G_i та значення функції у витоках

Підграф	Код витоку	Значення $f(x)$
G_6	213456	126
G_5	213465	125
G_4	213564	123
G_3	214563	119
G_2	314562	113
G_1	324561	108

З таблиці видно, що підграф G_1 можна не розглядати, так як максимальне значення функції менше того значення, яке ми шукаємо. Отже, розглянемо інші підграфи, наприклад G_6 і дослідимо в ньому п'яту координату. Вона послідовно приймає значення $5 \Rightarrow 4 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$. Відповідно до вектора коефіцієнтів функції $c = (2,3,4,6,7,8)$ це приведе до зменшення значення функції послідовно на $(c_5 - c_4) = 1,$ $(c_5 - c_4) = 3,$ $(c_5 - c_1) = 5,$ $(c_5 - c_2) = 4,$ а в сумі на 13. Тобто найменше значення при $x_4 = 4$ функція приймає у вершині (324516) а саме $126 - 13 = 113.$ Треба зменшувати значення x_4. Це досягається транспозицією елементів 5 і 4 та зменшенням значення функції на $(c_4 - c_3) = 2,$ і воно стане рівним111. Але це все одно більше 109, тому знову замінюємо значення x_4 на 3, що зменшить значення функції на $(c_5 - c_2) = 4,$ тобто до 107, у вершині (425316). Тепер це менше 109, тому треба збільшувати x_5 до 2, що досягається транспозицією чисел 2 та 1 та зростанням функції у вершині (415326) на $(c_4 - c_1) = 4,$ тобто до 111. Знову зменшуємо значення x_4 до 1 шляхом транспозиції чисел 3 та 1. У вершині (435126) отримаємо зменшення функції на $(3-1)(c_4 - c_2) = 6,$ тобто до 105. Необхідно збільшити x_5 до 3, переставляючи числа 2 і 3. Отримаємо значення функції $105 + (c_5 - c_2) = 109,$ що і було потрібно. Оскільки x_4 в цій вершині найменше, то більше в підграфі G_6 таких вершин не існує. Переходимо до $i < 6$.

Зробимо деякі узагальнення. Розв'язання задачі провадиться для всіх типів вершин. Для фіксованого типу вершин послідовно для кожного підграфу знаходяться необхідні вершини x^*, в яких $f(x^*) = y^*$. Будемо далі називати вершину-виток початковою вершиною графа G_i. Алгоритм пошуку необхідної вершини для підграфу $G_i \, (i = 1, 2, \ldots, n)$ можна поділити на три етапи.

Перший етап:побудова коду початкової вершини для G_i.

Другий етап:розгортання графа вздовж координати x_4.

Третій етап: розгортання графа вздовж координати x_5.

Опишемо кожний етап.

I. Нехай вибрано тип вершини $\left(i_1, i_2, i_3\right)$, де $i_1 \cup i_2 \cup i_3 = \left\{1, 2, 3\right\}$, та номер підграфу *i*. Тоді покладемо: $x_6 = i;$ $x_5 = max\left\{N_6 \setminus x_6\right\}$, $x_4 = max\left\{N_6 \setminus \left(x_5, x_6\right)\right\}$. Упорядкуємо числа $\left\{N_6 \setminus \left(x_4, x_5, x_6\right)\right\}$ по зростанню $j_1 < j_2 < j_3$. Тоді $x_1 = j_{i_1}$, $x_2 = j_{i_2}$, $x_3 = j_{i_3}$. Це і буде код головної виршини мережі G_i, який позначимо p_1 (або q_1). Обчислимо значення цільової функції в цій вершині $f\left(p_1\right)$.

II. Розглянемо в цьому коді значення $x_k\left(k = 1, 2, 3, 4\right)$ та упорядкуємо їх за спаданням $x_4 = j_4 > j_3 > j_2 > j_1$. Розгортанням графа вздовж координати x_4 (або вниз) назвемо послідовність транспозицій $j_4 \Leftrightarrow j_3 \Leftrightarrow j_2 \Leftrightarrow j_1$, які крім коду головної вершини приводять до створення ще трьох кодів, які ми позначимо p_2, p_3 та p_4 і запишемо один під одним. Щоб знайти значення функції на цих перестановках, не обов'язково використовувати всі її координати. Достатньо знайти різницю значень функцій у сусідніх вершинах. Позначимо $\mu\left(\lambda\right)$ номер місця числа j_λ в коді перестановки $p_1\left(\lambda = 1, 2, 3\right)$. Тоді значення $f\left(p_2\right)$ буде менше $f\left(p_1\right)$ на величину $\Delta_1 = \left(j_4 - j_3\right)\left(c_4 - c_{\mu(3)}\right)$. У другому множнику постійно буде величина c_4, тому що в транспозиції завжди бере участь координата x_4. Аналогічно знаходимо другу та третю різниці. Очевидно, що в процесі подальшого пошуку необхідно використовувати тільки ті перестановки, для яких $f\left(p_1\right) \geq y^*$.

III. Розглянемо в коді вершини p_k (яку позначимо q_1, $\left(\kappa = 1, 2, 3, 4\right)$ значення x_5, x_4, x_3, x_2, x_1 та упорядкуємо їх за спаданням $x_5 = j_5 > j_4 > j_3 > j_2 > j_1$. Розгортанням графа вздовж координати x_5 (або праворуч) назвемо послідовність транспозицій $j_5 \Leftrightarrow j_4 \Leftrightarrow j_3 \Leftrightarrow j_2 \Leftrightarrow j_1$, які крім коду головної вершини повинні привести до створення ще чотирьох кодів, які ми позначимо q_2, q_3, q_4 та q_5. Але ці коди створювати не обов'язково. Так само, як і при розгортанні графа вниз, значення функції $f(q_2)$ у вершині q_2 буде меншим від значення $f(q_1)$ на величину $\delta_1 = (j_5 - j_4)(c_5 - c_{\mu(4)})$. За цією формулою знаходимо всі інші різниці. Послідовно віднімаючи $\delta_\lambda\ (4 \geq \lambda \geq 1)$ від $f(q_1)$, можемо отримати дві такі ситуації:

1. Всі значення функцій більші y^*. В цьому випадку переходимо до розгортання наступного коду p_k.

2. На деякому кроці λ^* отримаємо значення функції у вершині рівне y^* (або менше y^*). В першому випадку запам'ятовуємо код відповідної вершини. Переходимо до розгортання наступного коду p_k, при цьому кількість кроків обмежується до $\lambda^* - 1$.

Після розгортання всіх $p_k\ (\kappa = 1, 2, 3, 4)$ пошук необхідних вершин підграфа G_i з фіксованим i закінчується. Проведемо необхідні обчислення для всіх підграфів, наведених в таблиці 4.2, використовуючи описані три етапи.

Для G_6 отримали вершину (425136), в якій $f(425136) = 109$.

Для G_5 $p_1 = q_1 = (213465)$, $f(q_1) = 125$, $(6 \Leftrightarrow 4 \Leftrightarrow 3 \Leftrightarrow 2 \Leftrightarrow 1)$.

Різниця значень пробігає числа $(6-4)(7-6) = 2$, $7-4=3$, $7-2=5$, $7-3=4$, в сумі = 14, тобто найменше значення функції при постійному значенні x_4 дорівнює $125-14=111>109$. Розгортаємо код p_2, для якого $f(214365) = 123$, $(6 \Leftrightarrow 4 \Leftrightarrow 3 \Leftrightarrow 2 \Leftrightarrow 1)$ і різниця значень пробігає числа $(6-4)(7-4) = 6, 1, 5, 4$, в сумі = 16. Значення функції спадають $123-117-116-$ $-111-107$. Як бачимо, тут немає шуканої вершини, тому переходимо до коду p_3, при цьому будемо розгортати код на крок менше. Це дає значення функції $f(p_3) = f(314265) = 119$, $(6 \Leftrightarrow 4 \Leftrightarrow 3 \Leftrightarrow 2)$. Різниця значень функції пробігає числа $(6-4)(7-4) = 6, 5, 1$, а значення функції спадають $119-113-$

$-108-107.$ Тут також немає шуканої вершини, тому переходимо до коду p_4, в якому перевіряємо тільки один крок $f\left(p_4\right)=f\left(324165\right)=116.$ Сусідня вершина відрізняється транспозицією чисел 6 і 4, а функція на 6 менша, або дорівнює $110<109.$ Отже в підграфі G_5 немає шуканих вершин.

Для G_4 $p_1=\left(213564\right),$ $f\left(p_1\right)=123,$ $(6\Leftrightarrow 5\Leftrightarrow 3\Leftrightarrow 2\Leftrightarrow 1).$ Різниця значень пробігає числа $\left(7-6\right)=1,$ $2\left(7-4\right)=6,$ $7-2=5,$ $7-3=4,$ що дає послідовність спадання функції $123-122-$ $-116-111-107.$ Переходимо до $p_2=\left(215364\right),$ $f\left(p_2\right)=119,$ а кроків робимо на один менше $(6\Leftrightarrow 5\Leftrightarrow 3\Leftrightarrow 2).$ Різниця значень пробігає числа $\left(7-4\right)=3,$ $2\left(7-6\right)=2,$ $7-2=5,$ а функція спадає $119-116-114-109.$

Це значення шуканої вершини, а її код легко обчислити – (316524) за трьома транспозиціями над кодом p_2. Переходимо до коду $p_3=\left(315264\right)$ і робимо два кроки – $f\left(p_3\right)=115,$ $(6\Leftrightarrow 5\Leftrightarrow 3).$ Це дає різниці $7-4=3,$ $2\left(7-2\right)=10,$ тобто спадання функції $112-102.$ Це означає, що при цьому заченні x_4 шуканої вершини немає. Переходимо до коду $p_4=\left(325164\right)$ і робимо один крок $(6\Leftrightarrow 5).$ Це дає різницю $7-4=3,$ або значення функції $112-3=109.$ Так що в підграфі G_4 отримаємо дві шуканих вершини.

Переходимо до G_3, де код $p_1=\left(214563\right),$ $f\left(p_1\right)=119,$ $(6\Leftrightarrow 5\Leftrightarrow 4\Leftrightarrow 2\Leftrightarrow 1).$ Різниці складають числа 1, 3, 10, 4. Спадання функції наступне – 119 – 118 – 115 – 105 – 101. Тут немає шуканої вершини, а в наступному коді робимо два кроки. $f\left(p_2\right)=117,$ а $p_2=\left(215463\right),$ $(6\Leftrightarrow 5\Leftrightarrow 4),$ що відповідає різницям 3, 1. При цьому функція спадає – 117 – 114 – 113. Для p_4 відразу отримуємо $f\left(p_3\right)=109,$ тому на цьому пошук в G_3 закінчується.

Залишилося зробити пошук в підграфі G_2. Код його головної вершини $p_1=\left(314562\right),$ $f\left(p_1\right)=113,$ $(6\Leftrightarrow 5\Leftrightarrow 4\Leftrightarrow 3\Leftrightarrow 1).$ Різниці складають числа 1, 3, 5, 8. Спадання функції наступне –

(113 – 112 – 109 – 104 – 96). Тут маємо одну шукану вершину $f\left(x^*\right)=y^*$, де $x^*=\left(315642\right)$. Переходимо до коду вершини $p_2=\left(315462\right)$, $f\left(p_2\right)=111$, робимо один крок ($6\Leftrightarrow 5$). Це приводить до транспозиції зі зменшенням функції до 108. На цьому і закінчиться пошук на підграфі G_2, а тим самим на підграфі, що відповідає типу вершини (213). В результаті для даного типу вершини було знайдено п'ять вершин, значення функції в яких дорівнює 109.

Схема пошуку необхідних вершин серед підграфів з фіксованим типом вершин не відрізняється від наведеної.

Слід зазначити такі відмінності алгоритму пошуку необхідних вершин координатним методом від горизонтального:

Перш за все, останній метод потребує менше обчислень. Необхідно обчислювати лише різниці значень функції, тоді як в поппередньому методі кожний раз необхідно обчислювати значення функції в новій вершині, при цьому кожен раз відтворювати код вершини. Крім того, останній метод дозволяє декілька модифікацій, пов'язаних з обчисленнях не тільки в кодах головних вершинах (витоках), а і симетричним підходом відносно вершин-стоків та іншими.

Цим і закінчується дослідження складних комбінаторних задач на множині перестановок. Розглянуто локалізацію лінійної функції комбінаторної задачі з використанням теорії графів і комбінаторних конфігурацій, запропоновано і реалізовано два принципових алгоритма методу локалізації значення лінійної функції на множині перестановок.

4.6. Встановлення гамільтоновості графів переставних многогранників для оптимізації лінійної функції

Розглянемо описаний граф переставного многогранника, враховуючи, що для значень цільової функції виконується співвідношення:

$$F\left(x\right)_{[A]}\geq F\left(x\right)_{[B]}\geq F\left(x\right)_{[C]}\geq F\left(x\right)_{[D]}.$$

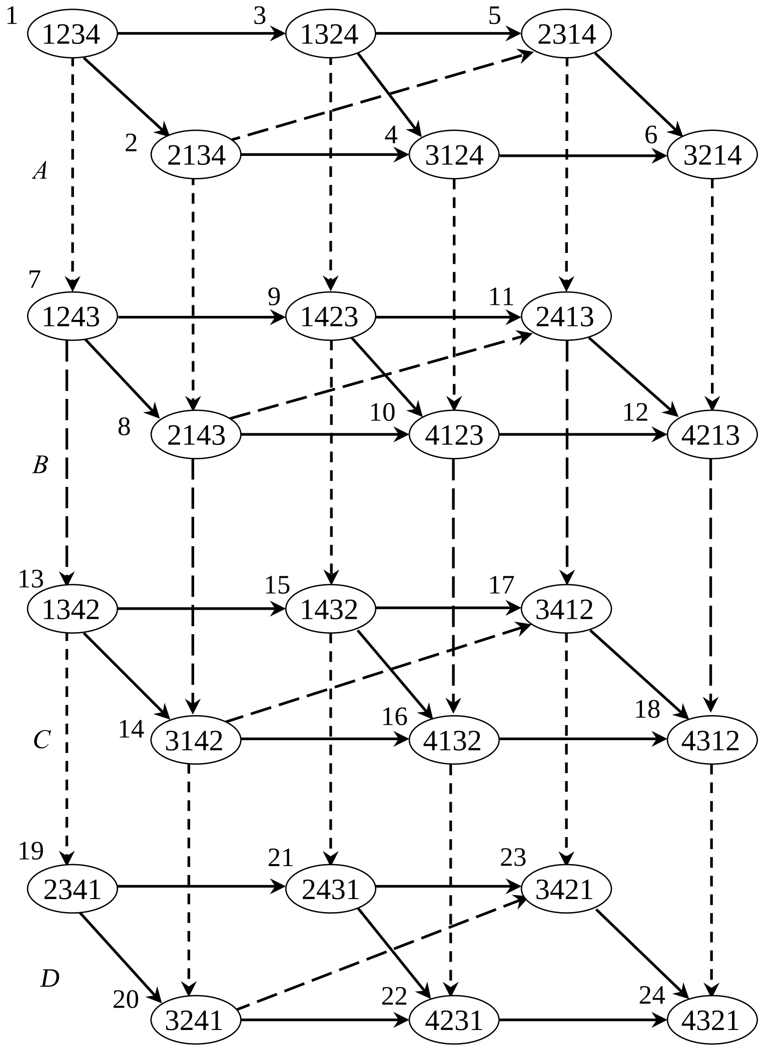


Рис. 4.20. Представлення графа перестановок $G(P_n)$ для $n = 4$

Вкажемо на основну властивість цього графа, що формулюються у вигляді наступних лем.

Лема 4.2. Елементи множини перестановок P_n можна розкласти на еквівалентні підграфи *A*, *B*, *C*, *D* у порядку спадання значень лінійної цільової функції $F(x)$ у вершинах графа $G(P_n) = A \bigcup B \bigcup C \bigcup D,$ в яких задається функція.

Лема 4.3. Розкладання точок комбінаторної конфігурації перестановок P_n при $n \geq 4$ забезпечує ієрархічне розташування цих точок по підграфах *A*, *B*, *C*, *D* (рис. 4.20) згідно значень заданої лінійної цільової функції $y^* = F\left(x^*\right)$.

Граф відображає часткову впорядкованість множини перестановок для $n = 4$ щодо значень довільної лінійної функції $f(x) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4$, де $c_1 \leq c_2 \leq c_3 \leq c_4$, а набір $x = (x_1, x_2, x_3, x_4)$ – пробігає множину всіх перестановок P_n. У графі дві вершини, що відповідають двом перестановкам, суміжні, якщо вони отримуються одна з іншої за допомогою транспозиції двох елементів і з'єднані одним ребром. Розглянемо дві суміжні вершини $p_1 = (x_1, \ldots, x_k, x_l, \ldots, x_n)$, $p_2 = (x_1, \ldots, x_l, x_k, \ldots, x_n)$.

Лема 4.4. З двох суміжних перестановок функція $f(x)$ приймає не менше (більше) значення для тієї, в якій максимальний з двох елементів, що розрізняються, знаходиться справа.

Ця лема справедлива для довільного *n*. Дійсно, нехай дано дві довільні перестановки $p_1 = (x_1, x_2, \ldots, x_k, \ldots, x_l, \ldots, x_n)$ і $p_2 = (x_1, x_2, \ldots, x_l, \ldots, x_k, \ldots, x_n)$, які відрізняються положенням двох елементів x_k і x_l, при цьому нехай $x_l > x_k$. Розглянемо різницю значень $f(p_1) - f(p_2)$. Підставивши значення координат перестановок в функцію $f(x)$, після спрощень одержимо, що вона дорівнює

$$c_k\left(x_k - x_l\right) + c_l\left(x_l - x_k\right) = \left(c_l - c_k\right)\left(x_l - x_k\right).$$

Оскільки для $l < k$ завжди $c_l \geq c_k$, при умові упорядкування коефіцієнтів цільової функції за зростанням, то цей вираз не менше нуля, що і підтверджує справедливість леми.

У графі на рис. 4.20 всі суміжні перестановки з'єднуються дугами відповідно лемі 4.4. Аналогічно будується граф переставного многогранника $G(P_n)$ для довільного *n*.

Наслідок 4.1. Максимальне значення лінійна функція $f(x)$ на графі перестановок $G(P_n)$ приймає в перестановці $(1,2,\ldots,n)$, а мінімальне – в перестановці $(n,n-1,\ldots,2,1)$.

Розглянемо задачі, які є актуальними для графів: знайти множину перестановок, в яких значення цільової функції дорівнює заданому значенню, тобто $x^* = \underset{x\in P_n}{arg}\, f(x)$, або визначити множину пар перестановок $(\underline{x},\overline{x})$, для яких при заданому y

$$\overline{x} = \underset{f(x)>y}{arg\,min}\, f(x), \quad \underline{x} = \underset{f(x)<y}{arg\,max}\, f(x). \qquad (4.11)$$

Підхід до розв'язання. Як і в попередньому випадку, використовуємо для розв'язування задачі граф, добудувавши на графі $G(P_n)$ множину дуг між несуміжними перестановками.

Множина дуг представляє шлях від початкової вершини, де функція $f(x)$ приймає максимальне значення, до кінцевої, де $f(x)$ приймає мінімальне значення.

Таким чином знову розглядається гамільтонів шлях, при цьому здійснюючи обхід по всім вершинам графа. Таким чином, визначається послідовність перестановок, через які він проходить, а методом дихотомії гамільтонового шляху, обчислюють значення функції у відповідній перестановці. Тоді завжди можна локалізувати довільне значення цільової функції $f(x)$.

Лема 4.5. Складність розв'язання задач (4.11), оцінюється зверху поліномом не вище за другий ступінь.

Оскільки число вершин графа $G(P_n)$ (перестановок) рівно $n!$, то складність обчислень при дихотомії оцінюється величиною $R = log_2\, n! = \sum_{i=2}^{n} log_2\, i$. Для її визначення розглянемо графік функції $y = log_2\, x$ на рис. 4.21.

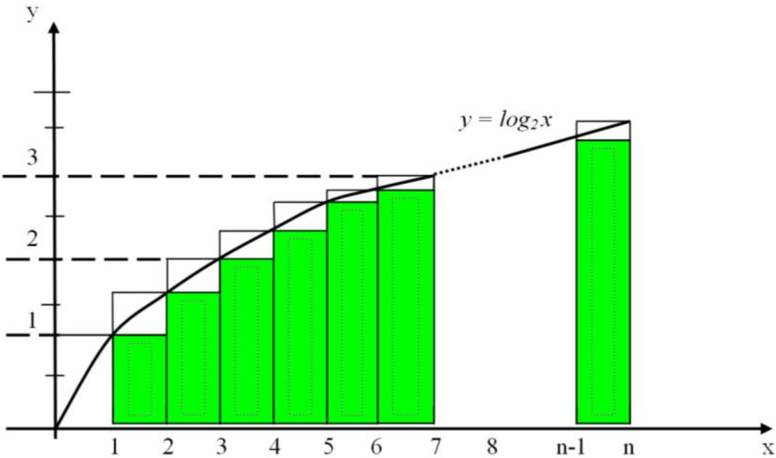


Рис. 4.21. Графік функції $y = log_2 x$

Площа всіх прямокутників, побудованих над кривою y, дорівнює $\overline{S} = \sum_{i=2}^{n} log_2 i = R$. Аналогічно, площа всіх прямокутників, побудованих під кривою y (заштрихованих), дорівнює $\underline{S} = \sum_{i=2}^{n-1} log_2 i = R - log_2 n$. З рисунка можна побачити, що площа, обмежена кривою $y = log_2 x$ і віссю абсцис, задовільняє обмеженням

$$R - log_2 n < \int_1^n log_2 x dx < R. \qquad (4.12)$$

Значення невизначеного інтегралу дорівнює $[x ln x - x] / ln 2$. Оскільки $n \geq log_2 n$, то одержимо оцінку $R \leq n^2$, що і потрібно довести.

Перейдемо тепер безпосередньо до побудови гамільтонового шляху в графі $G(P_4)$. Для зручності операції над перестановками пронумеруємо їх від 1 до 24, як показано на рис. 4.20, і будемо їх називати точками (вершинами) $p_i\left(i=1,2,\ldots,24\right)$, наприклад $p_{16}=\left(4,1,3,2\right)$. Вектор коефіцієнтів функції $f\left(x\right)$ позначимо $c=\left(c_1,c_2,\ldots,c_n\right)$, на рис. 4.1 це $c=\left(c_1,c_2,c_3,c_4\right)$. Тоді значення функції $f\left(x\right)$ у довільній точці $p_i\left(1\le i\le 24\right)$ визначається як скалярний добуток $f\left(p_i\right)=\left(p_i,c\right)$. Покажемо, що граф $G\left(P_4\right)$ можна побудувати індуктивним способом, починаючи з двох перших перестановок. Уявимо, що вершини p_1 і p_2 складають підграф графа $G\left(P_4\right)$, у якого фіксовані два останні елементи 3 і 4. Вершина p_2 утворена з p_1 шляхом транспозиції елементів 1 і 2, тому згідно лемі 4.1 $f\left(p_1\right)\ge f\left(p_2\right)$. Якщо тепер у вершинах p_1 і p_2 поміняти місцями елементи 2 і 3, то одержимо вершини p_3 і p_4, у яких залишається співвідношення $f\left(p_3\right)\ge f\left(p_4\right)$, крім того, за лемою 4.1 одержуємо $f\left(p_1\right)\ge f\left(p_3\right)$ і $f\left(p_2\right)\ge f\left(p_4\right)$. Аналогічно, шляхом транспозиції елементів 1 і 2 одержимо з p_3 і p_4 вершини p_5 і p_6. В результаті цих дій одержимо підграф A, який містить всі перестановки P_4 з фіксованим четвертим елементом $x_4=4$. Очевидно, що в підграфі A $f\left(x\right)$ приймає максимальне значення у вершині p_1 і мінімальне – у вершині p_6. Проте для побудови гамільтонового шляху в цьому підграфі дуг не вистачає. Можна було б побудувати ще одну дугу з вершини p_2 до вершини p_5, оскільки ці перестановки відрізняються між собою транспозицією чисел 1 і 3, але цього також не вистачає (ця дуга відмічена пунктиром).

Візьмемо тепер всі перестановки підграфа A і одночасно зробимо в них транспозицію чисел 3 і 4. В результаті одержимо підграф B, який містить всі ті перестановки P_4, у яких фіксований четвертий елемент $x_4=3$. За лемою 4.1 відповідні вершини підграфів A і B суміжні і з’єднуючі їх дуги йдуть зверху (від підграфа *A*) вниз (до підграфа *B*). Очевидно, що внутрішня орієнтація підграфа *B* зберігає внутрішню орієнтацію підграфа *A*. Якщо тепер у підграфі *B* у всіх перестановках зробити транспозицію чисел 2 і 3, то в результаті одержимо підграф *C*, що містить всі ті перестановки P_4, у яких зафіксований четвертий елемент $x_4 = 2$. Очевидно, що цей підграф також зберігає внутрішню орієнтацію підграфа *B* (а також підграфа *A*), і в кожну його вершину входить дуга від відповідної вершини підграфа *B*. І, нарешті, якщо в підграфі *C* у кожній перестановці зробимо транспозицію чисел 1 і 2, то одержимо підграф *D*, який містить всі перестановки P_4 з фіксованим четвертим елементом $x_4 = 1$.

Всі чотири підграфи A, B, C, D утворюють граф $G\left(P_4\right)$. Можна говорити, що він побудований індуктивним шляхом, починаючи з двох вершин p_1 і p_2. Спочатку ці вершини як би проектувалися послідовно двічі, утворюючи підграф *A*. Потім, підграф *A* проектувався тричі, утворюючи весь граф $G(P_4)$.

Повернемося тепер до питання про побудову гамільтонового шляху в підграфі *A*. Було б принципово важливим встановити співвідношення значень функцій в точках p_2 і p_3, а також в точках p_4 і p_5, Розглянемо відповідні різниці, які позначимо δf :

$$\delta f\left(2{,}3\right) = f\left(p_2\right) - f\left(p_3\right) = \left(p_2, c\right) - \left(p_3, c\right) = c_1 - 2c_2 + c_3;$$

$$\delta f\left(4{,}5\right) = f\left(p_4\right) - f\left(p_5\right) = \left(p_4, c\right) - \left(p_5, c\right) = c_1 - 2c_2 + c_3. \quad (4.13)$$

Ці різниці співпадають, тому якщо ми знаємо значення $\delta f\left(2{,}3\right)$, то цього достатньо для визначення гамільтонова шляху в підграфі *A*. Якщо $\delta f\left(2{,}3\right) \geq 0$, то одержуємо дуги $p_2 p_3$ і $p_4 p_5$, що визначає гамільтонов шлях $\sigma_1\left(A\right) = \left(1, 2, 3, 4, 5, 6\right)$. Якщо $\delta f\left(2{,}3\right) < 0$, то одержуємо дуги $p_3 p_2$ і $p_5 p_4$, що дає гамільтонов шлях $\sigma_2\left(A\right) = \left(1, 3, 2, 5, 4, 6\right)$.

З огляду на те, що при проектуванні підграфа *A* на підграфи *B, C* і *D* перші три елементи в перестановках міняються, то співвідношення (4.13), справедливі для підграфа *A*, на подальші підграфи не переносяться. Введемо позначення: $\Delta_i = c_{i+1} - c_i \left(i = 1, 2, \ldots, n-1 \right)$. Нехай $\alpha\left(\alpha_1, \alpha_2, \ldots, \alpha_{n-2}\right)$ – вектор, що відображає особливості заданої лінійної функції:

$$\Delta_i = \alpha_{i-1}\Delta_{i-1}\left(i = 2, 3, \ldots, n-1\right).$$

У кожному підграфі A, B, C, D четвертий елемент всіх перестановок фіксований, тому структура цих підграфів залежить тільки від перших трьох елементів, які для загального випадку позначимо $\left(i_1, i_2, i_3\right)$, де $\left(i_1 < i_2 < i_3\right)$. У кожному підграфі ці елементи визначають першу вершину p_j, де $j \equiv 1\left(mod\, 6\right)$, тобто вершини p_1, p_7, p_{13}, p_{19}. Якщо для довільного *n* з $G\left(P_n\right)$ вибрати підграф, у якого в перестановках зафіксовані $n-4$ останні елементи $\left(i_5, i_6, \ldots, i_{n-1}, i_n\right) \in N_n$, то одержимо підграф типу $G\left(P_4\right)$, у якого замість елементів 1, 2, 3, 4 будуть елементи $i_1 < i_2 < i_3 < i_4$. У графі *A* буде зафіксований четвертий елемент i_4, в графі *B* – елемент i_3, в графі *C* – елемент i_2, і в графі *D* – елемент i_1.

Визначення 4.2. Назвемо структурними коефіцієнтами графа $G\left(P_4\right)$ величини:

$$\lambda_A = \frac{i_2 - i_1}{i_3 - i_2};\ \lambda_B = \frac{i_2 - i_1}{i_4 - i_2};\ \lambda_C = \frac{i_3 - i_1}{i_4 - i_3};\ \lambda_D = \frac{i_3 - i_2}{i_4 - i_3}.$$

Для побудови гамільтонового шляху в кожному підграфі необхідно обчислити різницю значень функції у вершинах p_i і p_j, де $i \equiv 2\left(mod\, 6\right)$, $j \equiv 3\left(mod\, 6\right)$, а також у вершинах p_k і p_l, де $k \equiv 4\left(mod\, 6\right)$, $l \equiv 5\left(mod\, 6\right)$. У підграфі *A* $p_i = \left(i_2, i_1, i_3, i_4\right)$, $p_j = \left(i_1, i_3, i_2, i_4\right)$. Тому

$$\delta f(2{,}3) = (p_{i,c}) - (p_{j,c}) = c_1(i_2 - i_1) - c_2(i_3 - i_1) + c_3(i_3 - i_2),$$

$$\delta f(2{,}3) = -(i_2 - i_1)\Delta_1 + (i_3 - i_2)\Delta_2. \qquad (4.14)$$

Дуга, що сполучає вершини p_2 і p_3, матиме напрям $p_2 p_3$, якщо $\delta f(2{,}3) \geq 0$. А це буде тільки при умові:

$$\frac{\Delta_2}{\Delta_1} \geq \frac{i_2 - i_1}{i_3 - i_2}, \text{ або } \alpha_1 \geq \lambda_A. \qquad (4.15)$$

Продовжуючи такі ж обчислення щодо вершин p_4 і p_5, одержимо $f(4{,}5) = -(i_3 - i_2)\Delta_1 + (i_2 - i_1)\Delta_2$. Дуга матиме напрям $p_4 p_5$, якщо виконуються умови

$$\frac{\Delta_2}{\Delta_1} \geq \frac{i_3 - i_2}{i_2 - i_1}, \text{або } \alpha_1 \geq {}^{1}\!/_{\lambda_A}. \qquad (4.16)$$

Аналогічні результати можна одержати і для інших підграфів. Наприклад, для підграфа C обчислимо $\delta f(14{,}15)$. Дуга матиме напрям $p_{14} p_{15}$, якщо

$$\frac{\Delta_2}{\Delta_1} \geq \frac{i_3 - i_1}{i_4 - i_3}, \text{ або } \alpha_1 \geq \lambda_C,$$

а дуга $p_{16} p_{17}$ матиме місце, якщо $\alpha_1 \geq {}^{1}\!/_{\lambda_C}$.

Розглянемо номери вершин графа $G(P_4)$ у класі лишків за $mod\, 6$, де за повну систему лишків прийнята множина (1, 2, 3, 4, 5, 6). Останні обчислення дають право стверджувати, що справедлива наступна теорема:

Теорема 4.2. У довільному підграфі X графа $G(P_4)$, де $X \in \{A, B, C, D\}$, гамільтонів шлях проходить в послідовності:

$$(1,3,2,5,4,6)(mod\, 6), \text{ якщо } \alpha_1 \leq min\left(\lambda_x, {}^{1}\!/_{\lambda_x}\right);$$

$$(1,2,3,4,5,6)(mod\, 6), \text{ якщо } \alpha_1 \geq max\left(\lambda_x, {}^{1}\!/_{\lambda_x}\right);$$

$$\left(1,3,2,4,5,6\right)\left(mod\,6\right),\text{ якщо } 1/\lambda_x \le \alpha_1 \le \lambda_x,\text{ і } \lambda_x \ge 1; \quad (4.19)$$

$$\left(1,2,3,5,4,6\right)\left(mod\,6\right),\text{ якщо } \lambda_x \le \alpha_1 \le 1/\lambda_x,\text{ і } \lambda_x \le 1.$$

Всі вказані чотири гамільтонові шляхи можуть мати місце при одній умові, якщо для a_1 ні в одному з обмежень (4.9) не виконується рівність. Якщо виконується рівність типу (4.7), то $f\left(p_i\right)=f\left(p_j\right)$, ребро $p_i p_j$ неорієнтоване і його можна проходити в довільному напрямі, звідки витікає, що шляхи перший і четвертий (або другий і третій) рівноправні. Якщо ж виконується рівність типу (8), то те ж справедливе щодо ребра $p_i p_j$, де $i \equiv 4\left(mod\,6\right)$, а $j \equiv 5\left(mod\,6\right)$, і тоді шляхи перший і третій (або другий і четвертий) рівноправні. Особливий випадок виникає при $\lambda_X = 1$. Якщо $\alpha_1 \ne 1$, то можливі тільки два перші гамільтонови шляхи. Якщо $\alpha_1 = 1$, то тоді всі чотири шляхи рівноправні, оскільки вказані дві пари вершин можна об'єднати і одержати дві вершини, а гамільтонові шляхи запишуться в загальному вигляді $\left[1,\left(2,3\right),\left(4,5\right),6\right]$ $\left(mod\,6\right)$. Щоб надалі уникати невизначеностей такого роду, значення α_1, що дорівнюють якому-небудь параметру λ_X (чи $1/\lambda_X$), приписуватимемо до лівого інтервалу. Тим самим ми доб'ємося, що в довільному випадку для кожного підграфа *A, B, C, D* буде вибраний один з чотирьох можливих гамільтонових шляхів. Чи означає це, що для побудови гамільтонового шляху в графі $G\left(P_4\right)$ доведеться розглядати 44 варіанти різних поєднань цих шляхів. Розглянемо це питання з урахуванням наступного твердження.

Лема 4.6. Для структурних коефіцієнтів підграфів A,B,C,D справедливі співвідношення:

(а) $\lambda_A > \lambda_B;\ \lambda_C > \lambda_B;\ \lambda_C > \lambda_D;$

(б) $1/\lambda_A < 1/\lambda_в;\ 1/\lambda_C < 1/\lambda_B;\ 1/\lambda_C < 1/\lambda_D.$

Очевидно, що (б) виходить з (а), тому достатньо довести справедливість (а). Перша нерівність виходить з того, що у двох величин чисельники однакові, а знаменник більше у λ_B, оскільки $i_4 > i_3$. Друга нерівність виходить з того, що у λ_C чисельник більше, а знаменник менше, ніж у λ_B. Третя нерівність витікає з того, що у двох величин знаменники однакові, а чисельник у λ_B більше, оскільки $i_3 - i_1$ більше $i_2 - i_1$.

Теорема 4.4. Існує не більш дев'яти сумісних варіантів побудови гамільтонових шляхів в підграфах A, B, C, D графа перестановок $G(P_4)$.

Лема 4.6 дозволяє встановити часткову впорядкованість для значень λ_X і $1/\lambda_X$, де $X \in \{A, B, C, D\}$. Ця залежність відображена на рис. 4.22

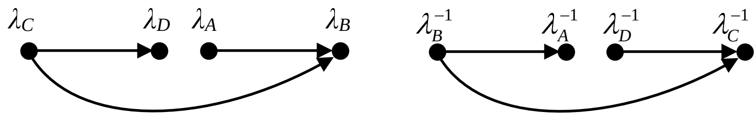


Рис. 4.22. Частково упорядковані структурні коефіцієнти

Якщо ніякі два з цих восьми параметрів не дорівнюють один одному, то при різних їх значеннях на числовій осі вони утворюють 9 інтервалів, в один з яких потрапляє конкретне значення α_1. Якщо деякі з цих параметрів співпадають, то число інтервалів зменшується, що і доводить теорему. Розглянемо приклад.

Приклад 4.4. Нехай $i_1 = 1, i_2 = 2, i_3 = 4, i_4 = 9.$ Знаходимо значення

$$\lambda_A = 1/2, \lambda_B = 1/7, \lambda_C = 3/5, \lambda_D = 2/5,$$

$$1/\lambda_A = 2,\ 1/\lambda_B = 7,\ 1/\lambda_C = 5/3,\ 1/\lambda_D = 5/2.$$

Після впорядкування одержимо зростаючу послідовність точок на числовій осі ($1/7,\ 2/5, 1/2, 3/5, 5/3, 5/2, 2, 7$).

Результати розрахунків наведені в таблиці 4.1.

Таблиці 4.1

Гамільтонові шляхи

№	Значення α_1	Гамільтонові шляхи в підграфах по вершинах			
		A	B	C	D
1	$\alpha_1 < 1/7$	1, 3, 2, 5, 4, 6	7, 9, 8, 11, 10, 12	13, 15, 14, 17, 16, 18	19, 21, 20, 23, 22, 24
2	$1/7 < \alpha_1 < 2/5$	1, 3, 2, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 15, 14, 17, 16, 18	19, 21, 20, 23, 22, 24
3	$2/5 < \alpha_1 < 1/2$	1, 3, 2, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 15, 14, 17, 16, 18	19, 20, 21, 23, 22, 24
4	$1/2 < \alpha_1 < 3/5$	1, 2, 3, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 15, 14, 17, 16, 18	19, 20, 21, 23, 22, 24
5	$3/5 < \alpha_1 < 5/3$	1, 2, 3, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 15, 14, 17, 16, 18	19, 20, 21, 23, 22, 24
6	$5/3 < \alpha_1 < 5/2$	1, 2, 3, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 14, 15, 17, 16, 18	19, 20, 21, 23, 22, 24
7	$5/2 < \alpha_1 < 2$	1, 2, 3, 5, 4, 6	7, 8, 9, 11, 10, 12	13, 14, 15, 16, 17, 18	19, 20, 21, 22, 23, 24
8	$2 < \alpha_1 < 7$	1, 2, 3, 4, 5, 6	7, 8, 9, 11, 10, 12	13, 14, 15, 16, 17, 18	19, 20, 21, 22, 23, 24
9	$7 < \alpha_1$	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 12	13, 14, 15, 16, 17, 18	19, 20, 21, 22, 23, 24

Неважко бачити, що всі гамільтонові шляхи в підграфах однозначно визначаються, якщо відомий тільки один параметр – α_1. Для побудови гамільтонового шляху на всьому графі $G(P_4)$ необхідно використовувати другий параметр – α_2. В залежності від поєднань цього параметра із значеннями структурних коефіцієнтів підграфів вийдуть різні варіанти гамільтонового шляху в графі $G(P_4)$. Якщо ці варіанти будуть відомі, то тоді задачі на перестановках для $n > 4$ можна зводити до підзадач на підграфах з $n = 4$.

Для $n = 5$, теорема так само виконується, оскільки в множині перестановок $P(A)$ розглядається $5! = 120$ точок, які розташовані на 5 гіперплощинах вигляду A (рис. 4.7) і містять по 24 точки кожна. Для розмірності n матимемо ту ж ситуацію, що і у попередньому випадку, тобто переставний многогранник завжди міститиме грані вигляду A, тому є необхідність розглядати α-питання.

Висновки до розділу

У даному розділі описується застосування нового методу направленого структурування для розв'язування екстремальних задач з лінійною цільовою функцією на комбінаторних конфігураціях перестановок.

Зокрема, в розділі розглянуто оптимізацію лінійної функції без додаткових обмежень, тобто комбінаторну безумовну задачу.

На підставі встановленого взаємозв'язку між задачами на комбінаторних множинах і графами комбінаторних конфігурацій вивчаються деякі структурні властивості допустимої області, а також сформульовано ряд тверджень, що дають можливість застосувати теорію графів для побудови методу розв'язання екстремальних задач.

Також в розділі вирішуються додаткові підзадачі, що використовуються в подальшому для реалізації методу направленого структурування. Розглянуто метод упорядкування значень лінійної функції на комбінаторній конфігурації перестановок, досліджено та побудовано орієнтований граф комбінаторної конфігурації перестановок та гамільтонів шлях в ньому за значеннями лінійної цільової функції.

На основі описаних в попередньому розділі нових методів генерування комбінаторних об'єктів побудовано алгоритм пошуку значень лінійної функції на лексикографічно упорядкованих перестановках.

Розглянуто алгоритм відображення номера перестановки N за елементом перестановки $P(A)$. Даний алгоритм дає можливість розв'язати обернену задачу: за заданою перестановкою знайти її номер в лексикографічно впорядкованій послідовності перестановок. Розглянуто реалізацію даного алгоритму на прикладах.

Розглянуто задачі на комбінаторній конфігурації перестановок та підхід до їх розв'язання, що ґрунтується на відповідності множини перестановок $P(A)$ повному орієнтованому графу, в якому перестановки представлені як вершини, а дві вершини $p_1, p_2 \in P_n(A)$, сполучені дугою, що йде від p_1 до p_2, якщо $F(p_1) \geq F(p_2)$. Доведено, що максимальне значення функція приймає у вершині, в яку не входить жодна дуга, а мінімальне – у вершині, з якої не виходить жодна дуга.

Побудовано гамільтонів шлях в такому графі, що відповідає послідовності перестановок, для яких значення функції утворює незростаючу послідовність. Передбачено, що тоді екстремальну задачу на комбінаторній конфігурації перестановок можна розв'язати шляхом дихотомії послідовності перестановок до тих пір, поки шуканий аргумент для заданого значення функції не потрапить в інтервал мінімальної довжини.

Встановлено умови гамільтоновості графів переставних многогранників для значення $n \leq 5$.

Досліджені та сформульовані властивості дають можливість побудови нових підходів до розв'язування інших типів екстремальних задач як умовних так і безумовних з іншими цільовими функціями та додатковими обмеженнями.

РОЗДІЛ 5. ОПТИМІЗАЦІЯ ДРОБОВО-ЛІНІЙНОЇ ФУНКЦІЇ НА КОМБІНАТОРНИХ КОНФІГУРАЦІЯХ

Значне розширення областей застосування обчислювальної техніки і детальніше використовування економіко-математичних моделей приводить до необхідності розв'язання складних задач великої вимірності. Великий інтерес представляють задачі, в яких цільова функція є дробово-лінійною [59, 97, 98, 278–280]. Дробово-лінійна функція, це функція, що характеризується відношенням двох лінійних форм. Як відомо, такі функції застосовуються в ряді прикладних задач оптимізації деяких відносних показників якості, таких як собівартість, рентабельність, продуктивність, трудомісткість і т.д. Моделі, що використовують вказані критерії, відображають тенденції постійного зниження рівня собівартості з розрахунку на одиницю продукції і підвищення якісних показників виробництва при збільшенні масштабів виробництва.

Дробово-лінійні критерії часто зустрічаються, наприклад, у області фінансової діяльності, при плануванні діяльності корпорацій, де необхідно мінімізувати відношення боргу до вартості власних засобів, максимізувати відношення випуску продукції на одного працюючого; у управління банківського балансу, де необхідно мінімізувати відношення ризикованих вкладень до капіталу, максимізувати відношення реального капіталу до необхідного капіталу, відношення застав на житлі до загальної суми застав і ін. Слід зазначити, що множини – область допустимих розв'язків задачі з дробово-лінійною функцією в багатьох задачах мають властивості комбінаторних конфігурацій: перестановок, розміщень, сполучень та ін.

5.1. Постановка задачі з дробово-лінійними функціями критеріїв на графах

Як було установлено в попередніх розділах, екстремальні задачі на комбінаторних конфігураціях представляють собою оптимізацію деякої функції $F(x)$ на комбінаторних конфігураціях, що визначаються множиною $A=\{\pi_i\}=\{(a_1,a_2,\ldots,a_r)\}$, де $\pi_i=a_1,a_2,\ldots,a_r$, за допомогою якої формуються перестановки, сполучення, розміщення, різні послідовності і т. п.

Екстремальні комбінаторні задачі з лінійною цільовою функцією тісно пов’язані з методами лінійного програмування. Якщо ж функція дробово-лінійна, то як правило застосовуються методи лінеаризації функції – тобто зведення її до лінійної форми. Далі розглядається задача знаходження екстремуму цієї функції на опуклій оболонці заданих точок – на опуклому многограннику. Знову екстремум лінійної форми на многограннику досягається в одній з вершин, які входять в множину даних елементів. Особливістю комбінаторних задач при такому зведенні залишиться те, що при знаходженні розв’язків необхідно визначити точки з цілочисловими координатами. Для розв’язання екстремальних задач з дробово-лінійною функцією це не завжди є доцільним і виправданим. Як було показано в попередніх розділах, навіть для лінійної цільової функції використання методів лінійного програмування не завжди адекватно дає можливість інтерпретувати розв’язки задачі. Але враховуючи особливості екстремальних задач на комбінаторних конфігураціях, для даної задачі з дробово-лінійною функцією цілі є також доцільним розглянути підзадачу визначення гамільтонового шляху, який відображає зміну значення дробово-лінійного функціонала на комбінаторних конфігураціях.

Розглядається задача на множині перестановок. Як відомо, переставний многогранник можна представити у вигляді графа, який описаний в роботі [87]. У цьому графі вершинами служить множина всіх перестановок P_n, а дві вершини $p_1, p_2 \in P_n$ утворюють дугу $\overrightarrow{p_1 p_2}$, якщо $f\left(p_1\right) \geq f\left(p_2\right)$, і якщо перестановка p_2 одержана з p_1 за допомогою транспозиції двох елементів.

Тоді розглянемо наступну **задачу:** знайти множину перестановок, в яких значення цільової функції рівно заданому значенню, тобто

$$x^* = \underset{x \in P_n}{arg}\ f\left(x\right),$$

де $f\left(x^*\right) = y$. Такі задачі розглянуті в [60, 61] для випадку, коли цільова функція – лінійна. У даному випадку функція представляється в дробово-лінійному вигляді, як відношення двох лінійних форм, що значно ускладнює задачу. Тоді вище сформульована задача зводиться до знаходження множини точок – перестановок, в яких досягається задане значення дробово-лінійного функціоналу:

$$x^{*} = \arg_{x\in P_n} F(x) = \arg_{x\in P_n} \frac{f(x)}{g(x)}, \tag{5.1}$$

де $y = F(x)$.

Також має сенс розглядати аналогічну задачу, де не завжди існують перестановки, в яких цільова функція приймає задане значення. Тоді вище сформульована проблема формулюється наступним чином: визначити множину пар перестановок $(\underline{x}, \overline{x})$, для яких при заданому y

$$\overline{x} = \arg\min_{F(x)>y} F(x) = \arg\min_{F(x)>y} \frac{f(x)}{g(x)},$$

$$\underline{x} = \arg\max_{F(x)<y} F(x) = \arg\max_{F(x)<y} \frac{f(x)}{g(x)}. \tag{5.2}$$

5.2. Властивості області допустимих значень дробово-лінійних функцій на графах

Зазначимо деякі властивості дробово-лінійної функції, які будуть корисні для подальшого розв’язання задачі.

Дробово-лінійна функція $F(x) = (\langle c, x\rangle + c_0) / (\langle d, x\rangle + d_0)$ не є ні увігнутою, ні опуклою. Поверхні рівня будь-якої функції $F(x)$, тобто множини $L_\alpha = \{x \in R^n | F(x) = \alpha\}$ є гіперплощинами.

Відомо, що будь-який локальний мінімум задачі дробово-лінійного програмування є в той же час глобальним, і якщо оптимальний розв’язок скінченний, то існує крайня точка многогранника M – області допустимих розв’язків, яка є оптимальною. Це твердження виконується, якщо чисельник і знаменник дробово-лінійної функції не перетворюються одночасно в нуль $\forall x \in X$.

Припустимо, що $\langle d, x\rangle + d_0 > 0 \; \forall x \in M$. Відомо, що на будь-якому прямолінійному відрізку, який належить многограннику *M*, дробово-лінійна функція $F(x)$ змінюється монотонно.

Теорема 5.1. Дробово-лінійна функція $F(x)$, досягає мінімуму (максимуму) тільки у вершинах многогранника *M*. Якщо мінімум (максимум) досягається в декількох крайніх точках, то він досягається і на їх опуклій оболонці.

Визначення 5.1. Неперервна функція $F(x)$ є квазіопуклою функцією на опуклій множині *M*, якщо виконується будь-яка з наступних еквівалентних умов:

(a) множина $\{x \in R^m \mid F(x) \le q, x \in M\}$ – опукла для всіх *q*, $M \subset R^n$;

(*b*) $x_1, x_2 \in M, F(x_2) < F(x_1) \Rightarrow F(\lambda x_2 + (1-\lambda)x_1) \le F(x_1)$, $0 \le \lambda \le 1$.

Визначення 5.2. Функція $F(x)$ є строго квазіопуклою на множині *M*, якщо в умові (*b*) фігурують строгі нерівності.

Теорема 5.2. Будь-який локальний мінімум в строго квазіопуклій функції є глобальним.

Оскільки множина

$$\left\{x \in R^n \,\middle|\, \left(\langle c\ , x\rangle + c_0\right) \big/ \langle d\ , x\rangle + d_0 \le q, x \in M\right\},$$

опукла для всіх значень *q*, то функція $F(x) = (\langle c, x\rangle + c_0) / (\langle d, x\rangle + d_0)$ є квазівипукла на множині *M*. Неважко показати, що функція $\langle c, x\rangle / \langle d, x\rangle$ є строго квазіопуклою на *M*.

Очевидно, що функція $F(x)$ є квазіувігнутою на опуклій множині *S*, якщо функція $(-F(x))$ квазіопукла.

Якщо зробити заміну: $F_i(x)$ на $-F_i(x)$, ту умову (*b*) для квазіувігнутих функцій можна записати в наступному вигляді:

$$F_i(x_2) > F_i(x_1) \Rightarrow F_i(\lambda x_1 + (1-\lambda)x_2) \ge F_i(x_1),$$

$$F_i\left(\lambda x_1+\left(1-\lambda\right)x_2\right)\ge F_i\left(x_1\right), \text{ або}$$

$$F_i\left(\lambda x_1+\left(1-\lambda\right)x_2\right)\ge min\left[F_i\left(x_1\right),F_i\left(x_2\right)\right],$$

для будь-яких $x_1, x_2 \in S$ і $0 \le \lambda \le 1$. Для увігнутих і квазіувігнутих функцій з точністю до зміни знаків зберігаються властивості опуклих і квазіопуклих функцій.

Як відомо, для розв'язання задач з дробово-лінійною функцією цілі існує велика кількість методів, які умовно поділяються на методи лінеаризації, параметричні методи, модифікації симплексу-методів, серед яких відомими є метод Чарнса і Купера, алгоритм Гілморі і Гоморі і інші. Як наголошується в [278, 279], всі ці методи мають близьку обчислювальну складність, а пріоритет віддається методу, що дозволяє ефективніше вирішувати дробово-лінійну задачу в конкретних практичних умовах (зважаючи на розмірність задачі, структура системи обмежень, наявність програмного забезпечення і т. п.). Найкращим є методи зведення до розв'язання серії задач лінійного програмування, для яких розроблено ефективне програмне забезпечення для різних типів комп'ютерів. Але, сьогодні представляють інтерес методи, в основі яких лежать структурні особливості області допустимих рішень, їх властивості. У основі розглядається підхід якраз і лежать деякі структурні властивості переставного многогранника і його графа, який є областю допустимих рішень задачі.

5.3. Підхід до розв'язання екстремальної задачі з дробово-лінійним функціоналом на перестановках

Розглянемо дробово-лінійну функцію $F\left(x\right)$, де чисельник $f\left(x\right)$ і знаменник $g\left(x\right)$ – дві лінійні функції, в яких коефіцієнти визначаються за правилом арифметичної прогресії:

$$c_i = c_1 + \Delta\left(i-1\right),\ d_i = d_1 + \delta\left(i-1\right). \qquad (5.3)$$

Згідно цим позначенням чисельник і знаменник дробово-лінійної функції для даної перестановки p можна представити відповідно: $f\left(x\right)=\left(\vec{c}, x\left(p\right)\right)$, де $x\left(p\right)$ – вектор змінних

$\left(x_{p(1)}, x_{p(2)}, \ldots, x_{p(n)}\right)$, а $g(x)=\left(\vec{d}, x(p)\right)$, де $p \in P_n$ – множина перестановок, а множина $\left(x_1, x_2, \ldots, x_n\right)$, де $x_i \in N$, тобто перестановці множини $(1, 2, \ldots, n)$ ставиться в відповідність вектор $p=\left(x_{p(1)}, x_{p(2)}, \ldots, x_{p(n)}\right)$.

Розглянемо значення дробово-лінійної функції в двох різних π, σ, перестановках, які відповідають номерам p_1, p_2 і порівняємо їх:

$$\frac{\left(\overline{c}, x(\pi)\right)}{\left(\overline{d}, x(\pi)\right)} - \frac{\left(\overline{c}, x(\sigma)\right)}{\left(\overline{d}, x(\sigma)\right)}.$$

Визначимо наступне співвідношення, в вигляді коефіцієнта $\rho(\pi, \sigma)$, що характеризуватиме деяку відмінність між значеннями функцій в сусідніх перестановках π, σ, що відрізняються одна від одної однією транспозицією:

$$\rho(p_1, p_2) = (p_1, \overline{c})(p_2, \overline{d}) - (p_2, \overline{c})(p_1, \overline{d}).$$

Тоді, очевидно що, якщо $\rho(p_1, p_2) \geq 0$, то $F(p_1) \geq F(p_2)$.

Розглянемо далі наступні міркування, підставивши значення змінних в чисельник і знаменник функції:

$$\overline{c} \cdot x(\pi) = c_1 x_{\pi(1)} + c_2 x_{\pi(2)} + \ldots + c_n x_{\pi(n)} = c_1 \sum_{i=1}^{n} x_i + \mathit{\Delta}\left(p_0 x(\pi)\right),$$

$p_0 = (0, 1, \ldots, n-1)$, $\sum_{i=1}^{n} x_i = S$, для знаменника аналогічно.

Визначимо співвідношення різниці:

$$\frac{c_1 S + \Delta\left(p_0 \cdot x(\pi)\right)}{d_1 S + \delta\left(p_0 \cdot x(\pi)\right)} - \frac{c_1 S + \Delta\left(p_0 \cdot x(\sigma)\right)}{d_1 S + \delta\left(p_0 \cdot x(\sigma)\right)},$$

зробимо математичні перетворення

$$c_1 d_1 S^2 + c_1 \delta S\left(p_0 x(\sigma)\right) + d_1 \Delta S\left(p_0 x(\pi)\right) +$$

$$-\Delta\delta\Big[\big(p_0\big(x(\pi)\big)\big)\cdot\big(p_0\big(x(\sigma)\big)\big)\Big]-$$

$$-c_1d_1S^2-c_1\delta S\big(p_0x(\pi)\big)-d_1\Delta S\big(p_0x(\sigma)\big)-$$

$$-\Delta\delta\Big[\big(p_0\big(x(\pi)\big)\big)\cdot\big(p_0\big(x(\sigma)\big)\big)\Big].$$

Зведемо подібні доданки, одержимо

$$S\left[c_1\delta-d_1\Delta\right]\big(p_0x(\sigma)\big)+S\left[c_1\delta-d_1\Delta\right]\big(p_0x(\pi)\big);$$

$$S\left[c_1\delta-d_1\Delta\right]\Big(p_0x(\sigma)-\big(p_0x(\pi)\big)\Big)=S\left[c_1\delta-d_1\Delta\right]p_0\big(x(\sigma)-x(\pi)\big).$$

Розглянемо дві перестановки:

$$\pi=\Big(x_{\pi(1)},x_{\pi(2)},\ldots,x_{\pi(j)},\ldots,x_{\pi(k)},\ldots\Big),$$

$$\sigma=\Big(x_{\pi(1)},x_{\pi(2)},\ldots,x_{\pi(k)},\ldots,x_{\pi(j)},\ldots\Big),$$

$$(j-1)\Big[-x_{\pi(j)}+x_{\pi(k)}\Big]+(k-1)\Big[x_{\pi(k)}-x_{\pi(j)}\Big]=$$

$$(k-j)\Big[x_{\pi(k)}-x_{\pi(j)}\Big].$$

Далі приведемо міркування для $n=3$, що визначає кількість елементів перестановки в множині P_n, складених із множини $A(a_1,a_2,a_3)$, оскільки їх же можна адаптувати на більш загальний випадок.

Для прикладу розглянемо дві перестановки $p_1=(1,2,3)$, $p_2=(1,3,2)$, підставивши їх в цільову функцію $F(x)$ і розглянемо знак різниці між ними:

$$\frac{c_1x_1+c_2x_2+c_3x_3}{d_1x_1+d_2x_2+d_3x_3}\sim\frac{c_1x_1+c_2x_3+c_3x_2}{d_1x_1+d_2x_3+d_3x_2}. \qquad (5.4)$$

Помітимо, що знак різниці визначається знаком чисельника, а знаменник не впливає на знак, тому розглянемо співвідношення чисельника (5.4).

Після елементарних математичних перетворень над пропорціями, одержимо наступний вираз:

$$x_2^2\left(c_2d_3-c_3d_2\right)+x_3^2\left(c_3d_2-c_2d_3\right)+$$
$$+x_1x_2\left(c_1d_3-c_3d_1+c_2d_1-c_1d_2\right)+$$
$$+x_1x_3\left(c_1d_2-c_2d_1+c_3d_1-c_1d_3\right). \qquad (5.5)$$

Введемо наступні позначення:

$$\begin{vmatrix} c_1, c_2 \\ d_1, d_2 \end{vmatrix} = m_{12}; \quad \begin{vmatrix} c_1, c_3 \\ d_1, d_3 \end{vmatrix} = m_{13}; \quad \begin{vmatrix} c_2, c_3 \\ d_2, d_3 \end{vmatrix} = m_{23}. \qquad (5.6)$$

Підставляючи позначення (5.6) в (5.5), одержимо наступний вирази:

$$x_2^2m_{23}-x_3^2m_{23}=m_{23}\left(x_2^2-x_3^2\right);$$
$$x_1x_2\left(c_1d_3-c_3d_1+c_2d_1-c_1d_2\right)=x_1x_2\left(m_{13}-m_{12}\right);$$
$$x_1x_3\left(c_1d_2-c_2d_1+c_3d_1-c_1d_3\right)=x_1x_3\left(m_{12}-m_{13}\right),$$

і після зведення подібних доданків, маємо:

$$m_{12}\left(x_1x_3-x_1x_2\right)+m_{13}\left(x_1x_2-x_1x_3\right)+m_{23}\left(x_2^2-x_3^2\right).$$

Введемо коефіцієнт $\rho\left(p_i,p_j\right)$, який визначає знак різниці між заданими перестановками p_i,p_j.

Визначення 5.3. Мінор матриці утвореної двома рядками коефіцієнтів, заданими згідно формули (5.3), цільової дробово-лінійної функції $F\left(x\right)$:

$$m_{ij}=\begin{vmatrix} c_i & c_j \\ d_i & d_j \end{vmatrix}$$

називається мінором цільової функції, що визначає знак між двома перестановками p_i,p_j у яких досягається значення цієї цільової функції.

Тоді, враховуючи (5.5) і правило знаходження визначника, між перестановками p_1, p_2 знак різниці визначається таким чином:

$$\rho\left(p_1, p_2\right)=\begin{vmatrix}1\,2\,3\\1\,3\,2\end{vmatrix}=m_{12}-m_{13}-5m_{23},$$

і залежить від мінору другого порядку m_{ij}.

Визначення 5.4. Назвемо коефіцієнти при мінорах m_{12}, m_{13}, m_{23} відповідно λ_{12}, λ_{13}, λ_{23}, тоді $\vec{\lambda}\left(i, j\right)=\left(\lambda_{12},\lambda_{13},\lambda_{23}\right)$.

Враховуючи, що для $n=3$ для лінійної функції $f\left(x\right)$ структурний граф переставного многогранника *M* матиме наступний вигляд:

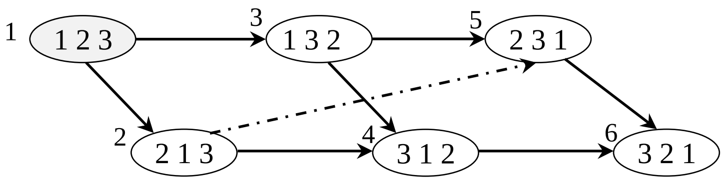


Рис. 5.1. Структурований граф переставного многогранника $n=3$

Визначимо відстані між точками: $\rho\left(1,3\right)=\left(1,-1,-5\right)$, $\rho\left(1,2\right)=\left(-3,-3,3\right)$, $\rho\left(2,4\right)=\left(-1,-5,-1\right)$, $\rho\left(3,4\right)=\left(-8,-4,4\right)$, $\rho\left(3,5\right)=\left(-3,-3,-3\right)$, $\rho\left(4,6\right)=\left(3,-3,-3\right)$, $\rho\left(5,6\right)=\left(-5,-1,1\right)$, $\rho\left(1,4\right)=\left(-5,-7,1\right)$, $\rho\left(1,6\right)=\left(-4,-8,-4\right)$, $\rho\left(2,3\right)=\left(5,1,-7\right)$.

Враховуючи, що коефіцієнти цільової функції визначені згідно наступних формул:

$$c_i=c_1+\Delta\left(i-1\right),\ d_i=d_1+\delta\left(i-1\right),$$

зробимо заміну у виразах (5.5), тоді маємо наступні вирази:

$$m_{12}=\begin{vmatrix} c_1, c_1+\Delta \\ d_1, d_1+\delta \end{vmatrix}=\delta c_1-\Delta d_1;$$

$$m_{13}=\begin{vmatrix} c_1, c_1+2\Delta \\ d_1, d_1+2\delta \end{vmatrix}=2\left(\delta c_1-\Delta d_1\right)=2m_{12};$$

$$m_{23}=\begin{vmatrix} c_1+\Delta, c_1+2\Delta \\ d_1+\delta, d_1+2\delta \end{vmatrix}=\begin{vmatrix} c_1+\Delta,\ \Delta \\ d_1+\delta,\ \delta \end{vmatrix}=\begin{vmatrix} c_1,\ \Delta \\ d_1,\ \delta \end{vmatrix}=m_{12}. \qquad (5.7)$$

Підставляючи (5.7) в значення відстаней між точками – перестановками, одержимо:

$$\rho(1,3)=-6m_{11},\ \ \rho(1,2)=-6m_{11},\ \ \rho(2,4)=-12m_{11},$$

$$\rho(3,4)=-12m_{11},\ \ \rho(3,5)=-6m_{11},\ \ \rho(4,6)=-6m_{11},$$

$$\rho(5,6)=-6m_{11}.$$

На підставі зроблених розрахунків коефіцієнтів можна побудувати гамільтонів шлях усередині кожної шестірки, елементів перестановки і прослідити зміни значень цільової функції у вершинах графа та на кожному з підграфів, так як і для лінійної функції.

На підставі вище висловлених міркувань для переставного многогранника $n=3$, сформулюємо деякі значущі факти для довільного значення n.

Лема 5.1. Для $n\geq 3$ і $(i\neq j)$ справедлива наступна рівність

$$m_{ij}=\begin{vmatrix} c_1+(j-i)\Delta \\ d_1+(j-i)\delta \end{vmatrix}=|i-j|\begin{vmatrix} c_1 & \Delta \\ d_1 & \delta \end{vmatrix},\ m_{ij}=|i-j|m_{12}.$$

Доведення леми виходить з вище висловлених міркувань.

Тоді граф перестановок при $n=4$ для значень дробово-лінійної функції має наступний вигляд, зображений на рис. 5.2.

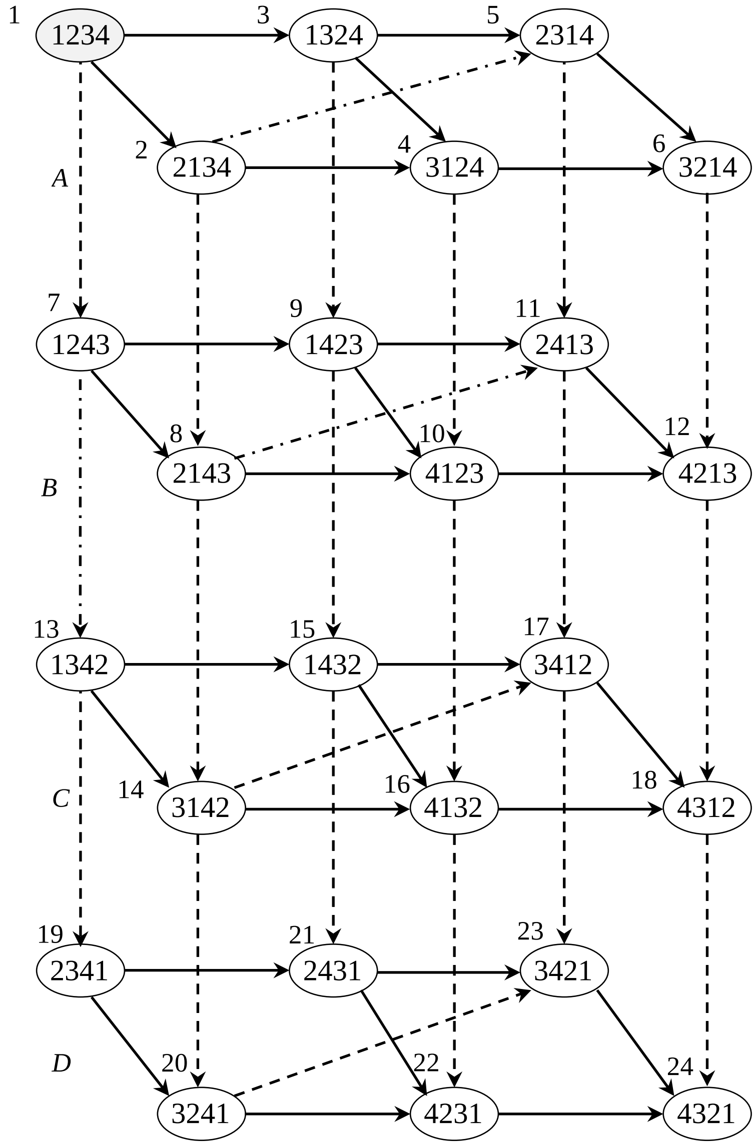


Рис. 5.2. Представлення графа перестановок $G(P_n)$ для $n = 4$

Теорема 5.3. Різниця між двома перестановками $\rho\left(p_i, p_j\right)$, $\left(i \neq j\right)$ для довільних n визначається по формулі

$$\rho\left(p_i, p_j\right) = \sum_{j=i+1}^{n} \sum_{i=1}^{n-1} \lambda_{ij} m_{ij}, \tag{5.8}$$

де λ_{ij} дорівнює мінору матриці $2 \times n$, утвореної двома перестановками p_i, p_j.

Доведення: проводимо методом по індукції:

1) для $k = 1$ цей вираз має місце;

2) для k цей вираз виконується на підставі вище висловлених міркувань;

3) для $k+1$ доведемо цей факт. Для цього розглянемо p_t, p_s, які утворені однією транспозицією, тобто відрізняються порядком слідування тільки однієї координати: $p_t = \left(i_1, i_2, \ldots, i_t, i_s, \ldots, i_k, i_{k+1}\right)$, $p_s = \left(i_1, i_2, \ldots, i_s, i_t, \ldots, i_k, i_{k+1}\right)$.

Складемо матрицю їх коефіцієнтів, яка має наступний вигляд:

$$\begin{bmatrix} i_1, i_2, \ldots, i_t, i_s, \ldots, i_k, i_{k+1} \\ \underbrace{i_1, i_2, \ldots,}_{t-1} i_t, i_s, \underbrace{\ldots, i_k, i_{k+1}}_{k+1-s} \end{bmatrix}. \tag{5.9}$$

Відповідно, щоб знайти різницю між двома перестановками $\rho\left(p_t, p_s\right)$, необхідно використати формулу (5.4). Далі знак різниці визначити перетвореннями формул (5.7). Формула (5.8) випливає із означення мінорів і коефіцієнтів при мінорах.

Теорема 5.4. Граф перестановок $\tilde{G}\left(P_n\right)$ для дробово-лінійної функції $F\left(x\right)$, коефіцієнти якої визначені згідно (5.2), співпадає з графом перестановок для лінійної функції $G\left(P_n\right)$, з точністю до орієнтації.

Доведення: випливає з леми 5.1.

Наслідок 5.1. Екстремальні значення функції $F\left(x\right)$, в якій коефіцієнти цільової функції визначаються по формулах (5.2), досягається в крайніх точках.

5.4. Чисельні приклади та аналіз експериментів

Розглянемо числові приклади, враховуючи, що для коефіцієнтів цільових функцій виконуються співвідношення (5.3).

Приклад 5.1. Знайти відношення двох лінійних функцій – значення дробово-лінійної функції, коли коефіцієнти функції побудовані по правилу арифметичної прогресії – в чисельнику збільшуються на 2, у знаменнику на 1. У таблиці 5.1 задані значення коефіцієнтів цільової функції, точки – вершини графа й значення цільової функції.

Таблиця 5.1

Зміна значень цільової функції

c_i	4	6	8
d_i	1	2	3

	x_1	x_2	x_3	$F(x)=\dfrac{(c_i, x_i)}{(d_i, x_i)}$
1	1	2	3	2,857143
2	2	1	3	2,923077
3	1	3	2	2,923077
4	3	1	2	3,090909
5	2	3	1	3,090909
6	3	2	1	3,2

На рисунку представлена динаміка зміни значення цільової функції.

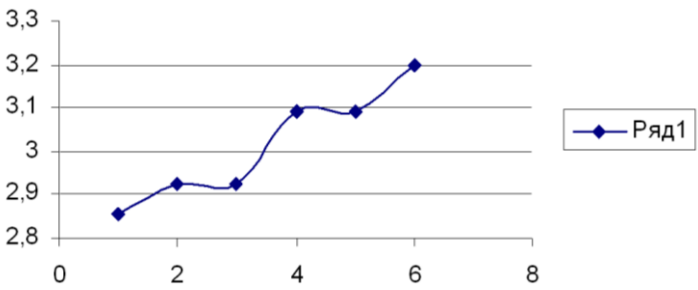


Рис. 5.3. Зміни значень дробно-лінійної функції

Приклад 5.2. Визначити відношення двох лінійних функцій − значення дробово-лінійної функції, коли коефіцієнти збільшуються в чисельнику на 2, у знаменнику на 1. У таблиці 5.2. задані значення коефіцієнтів цільової функції, точки − вершини графа й значення цільової функції.

Таблиця 5.2

Зміна значень цільової функції

c_i	5	7	9
d_i	3	4	5

	x_1	x_2	x_3	$F(x)=\dfrac{(c_i, x_i)}{(d_i, x_i)}$
1	1	2	3	1,769231
2	2	1	3	1,76
3	1	3	2	1,76
4	3	1	2	1,73913
5	2	3	1	1,73913
6	3	2	1	1,727273

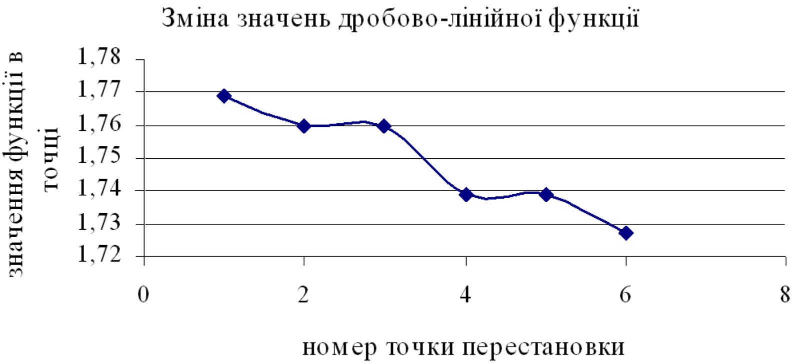


Рис. 5.4. Зміна значень дробово-лінійної функції

Приклад 5.3. Знайти відношення двох лінійних функцій – значення дробово-лінійної функції, де коефіцієнти змінюються в чисельнику на +2, у знаменнику – на +1. У таблиці 5.3 задані значення коефіцієнтів цільової функції, точки – вершини графа й значення цільової функції при $n = 4$.

Таблиця 5.3

Зміна значень цільової функції

c_i	5	7	9	11
d_i	3	4	5	7

№ п/п	x_1	x_2	x_3	x_4	$F(x) = \dfrac{(c_i, x_i)}{(d_i, x_i)}$
1	1	2	3	4	1,66666667
2	2	1	3	4	1,66037736
3	1	3	2	4	1,66037736
4	3	1	2	4	1,64705882
5	2	3	1	4	1,64705882
6	3	2	1	4	1,64
7	1	2	4	3	1,69230769
8	2	1	4	3	1,68627451
9	1	4	2	3	1,68
10	4	1	2	3	1,65957447
11	2	4	1	3	1,66666667
12	4	2	1	3	1,65217391
13	1	3	4	2	1,71428571
14	3	1	4	2	1,70212766
15	1	4	3	2	1,70833333
16	4	1	3	2	1,68888889
17	3	4	1	2	1,68181818
18	4	3	1	2	1,6744186
19	2	3	4	1	1,73333333
20	3	2	4	1	1,72727273
21	2	4	3	1	1,72727273
22	4	2	3	1	1,71428571
23	3	4	2	1	1,71428571
24	4	3	2	1	1,70731707

На графіку рис. 5.5 представлена тенденція зміни значень цільової функції, що має поліноміальну залежність.

Якщо проаналізувати всі графіки, то слід зазначити, що розташування точок підтверджує сформульовані теореми і їх обґрунтування.

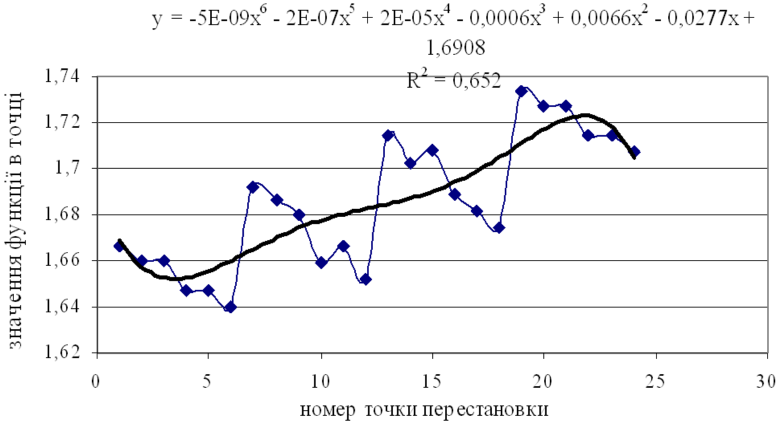


Рис. 5.5. Зміна значень дробово-лінійної функції

Продовжуючи й розвиваючи результати цієї роботи, варто зазначити, що має сенс розглядати задачу, де не завжди існують перестановки, у яких цільова функція приймає задане значення. Тоді вище сформульована проблема стане як **задача:** визначити множину пар перестановок $\left(\underline{x}, \overline{x}\right)$, для яких при заданому y

$$\overline{x} = \underset{F(x)>y}{arg\,min}\, F\left(x\right) = \underset{F(x)>y}{arg\,min} \frac{f\left(x\right)}{g\left(x\right)},$$

$$\underline{x} = \underset{F(x)<y}{arg\,max}\, F\left(x\right) = \underset{F(x)<y}{arg\,max} \frac{f\left(x\right)}{g\left(x\right)}.$$

Модель задачі з урахуванням комбінаторних властивостей області допустимих розв’язків і дробово-лінійних функцій критеріїв може бути успішно застосована при розв’язанні різних практичних задач. Установлено взаємозв’язок між задачею на комбінаторній множині перестановок і задачею на неперервній допустимій множині, а також обґрунтоване використання й застосування теорії графів для побудови методу розв’язання сформульованої задачі. На підставі доведених теорем, продовжуючи дослідження й розвиваючи результати попередніх робіт авторів, запропонований підхід до розв’язання багатокритеріальної задачі з дробово-лінійними критеріями на допустимій множині комбінаторної конструкції перестановок, що полягає у зведенні пошуку розв’язку вихідної задачі до задачі побудови гамільтонового шляху на графі перестановок.

5.5. Дослідження екстремальної дробово-лінійної задачі при умові кусково-лінійної тенденції знаменника

У даному пункті розглядається екстремальна задача з дробово-лінійною функцією критеріїв на перестановках, у якій знаменник являє собою кусково-лінійну тенденцію, досліджено властивості області допустимих розв’язків для такої задачі, побудовано метод розв’язання.

Функція представляється в дробово-лінійному вигляді, як відношення двох лінійних форм за умови, що знаменник або чисельник функції можуть бути кусково-лінійними, що значно ускладнює задачу. Тоді задача зводиться до знаходження множини точок − перестановок, у яких досягається задане значення дробово-лінійного функціоналу:

$$x^* = \underset{x \in P_n}{arg}\, F(x) = \underset{x \in P_n}{arg}\, \frac{f(x)}{g(x)},$$

де $y = F(x)$.

Розглянемо дробово-лінійну функцію $F(x)$, де чисельник $f(x)$ і знаменник $g(x)$ – дві лінійні функції, у яких коефіцієнти визначаються за правилом арифметичної прогресії, як в попередньому випадку:

$$c_i = c_1 + \Delta(i-1);$$

$$d_i = d_1 + \sigma(n-1). \qquad (5.10)$$

Згідно із цими позначеннями чисельник і знаменник дробово-лінійної функції можна представити відповідно:

$$f(x) = (\vec{c}, \pi(x)), \;\; g(x) = (\vec{d}, \pi(x)),$$

де $\pi(x) = \left(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}\right)$, а $\bar{c} = (c_1, c_2, \ldots, c_n)$, $c_1 \le c_2 \le \ldots \le c_n$ – коефіцієнти цільової функції.

Розглянемо значення дробово-лінійної функції у двох різних перестановках і зрівняємо їх: $F(\pi) - F(\sigma)$.

З огляду на властивості дробово-лінійної функції, а вона розглядається як кусково-лінійна в знаменнику, одержимо:

$$(\vec{c} \cdot \pi(x)) = c_1 \sum_{i=1}^{n} x_i + \Delta \sum_{i=2}^{n} (i-1) x_{\pi(i)};$$

$$(\vec{c} \cdot \sigma(x)) = c_1 \sum_{i=1}^{n} x_i + \Sigma \sum_{i=2}^{n} (i-1) x_{\sigma(i)}.$$

Позначимо $S = \sum_{i=1}^{n} x_i$; $u_0 = (0, 1, \ldots, n-1)$. Тоді

$$f(\pi(x)) = Sc_1 + \Delta(u_0 \cdot \pi(x)); \;\; f(\sigma(x)) = Sc_1 + \Delta(u_0 \cdot \sigma(x)).$$

Аналогічно

$$g(\pi(x)) = Sd_1 + \delta(u_0 \cdot \pi(x)); \;\; g(\sigma(x)) = Sd_1 + \delta(u_0 \cdot \sigma(x)).$$

З огляду на вище записані співвідношення, формула (5.8) прийме вигляд:

$$F(\pi) - F(\sigma) \approx f(\pi(x)) \cdot g(\sigma(x)) - f(\sigma(x)) g(\pi(x)) =$$

$$= S^2 c_1 d_1 + Sc_1 \cdot \delta(u_0 \cdot \sigma(x)) + Sd_1 \Delta(u_0 \cdot \pi(x)) +$$

$$+\Delta\delta\left(u_0\cdot\pi(x)\right)\left(u_0\cdot\sigma(x)\right)-S^2c_1d_1-Sc_1\delta\left(u_0\cdot\pi(x)\right)-$$

$$-Sd_1\Delta\left(u_0\cdot\sigma(x)\right)-\delta\Delta\left(u_0\cdot\sigma(x)\right)\left(u_0\cdot\pi(x)\right)=$$

$$=Sc_1\sigma\cdot u_0\left[\sigma(x)-\pi(x)\right]+Sd_1\Delta\cdot u_0\left[\pi(x)-\sigma(x)\right]=$$

$$=Su_0\left[\pi(x)-\sigma(x)\right]\left(d_1\Delta-c_1\delta\right).$$

Оскільки $u_0\left[\pi(x)-\sigma(x)\right]=(l-k)\left(x_e-x_k\right)\geq 0,$ то знак ΔF залежить від різниці $d_1\Delta-c_1\delta=\begin{vmatrix}c_1 & d_1\\ \Delta & \delta\end{vmatrix}.$

Якщо $\Delta F\geq 0,$ то граф $G\left(P_4\right)$ збігається з аналогічним графом для лінійної функції. Якщо $\Delta F\leq 0,$ то граф $G\left(P_4\right)$ має протилежну орієнтацію.

Нехай тепер $f(x)=\left(\vec{c}x\cdot\pi(x)\right),$ де $c_i=c_1+(c-1)\Delta,(\Delta>0),$ а знаменник являє собою кусочно-лінійну функцію $g(x)=\left(\vec{d}\cdot\pi(x)\right),$ тоді

$$\left.\begin{cases}d_i=d_1+\delta_1(i-1) & i\leq k\\ d_i=d_k+\delta_2(i-k) & i\geq k\end{cases}\right\}\quad\begin{pmatrix}\delta_1>0\\ \delta_2>\delta_1\end{pmatrix}.$$

Тоді $f\left(\pi(x)\right)=S_1c_1+\Delta\left(u_0\cdot\pi(x)\right).$

З огляду на особливості знаменника, для $g(x)$ необхідно перестановки розбити на дві частини:

$$x=\left(x_1,x_2,\ldots,x_n\right)=X_1\cup X_2,$$

де $X_1=\left(x_1,x_2,\ldots,x_k\right),\ X_2=\left(x_{k+1},x_{k+2},\ldots,x_n\right).$

Тоді $\sum\limits_{i=1}^{k}x_i=S_1,\ \ \sum\limits_{i=k+1}^{n}x_i=S_2\cdot p_1,p_2\in P_n.$

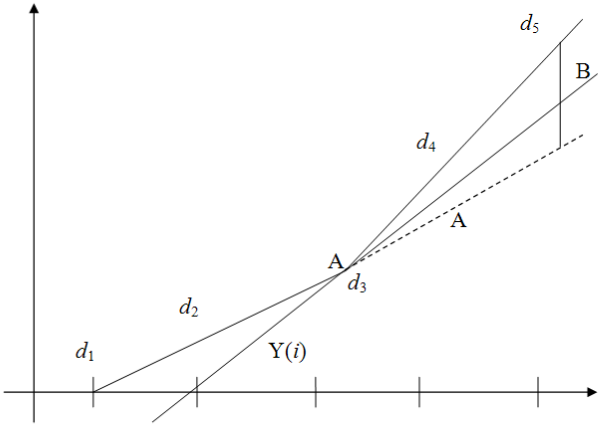


Рис. 5.6. Побудова кусково-лінійної функції знаменника

Згідно рис. 5.6 знаменник складається із двох шматків, тому необхідно побудувати графік функції у вигляді двох частин. Порядок побудови буде наступним:

1) розбити кусково-лінійну функцію знаменника на дві частини, розділивши точкою А; тоді функція знаменника прийме вигляд

$$g(x) = Y(i) + |i - k| \cdot \lambda + d_1; \tag{5.11}$$

2) необхідно зробити продовження першого (нижнього шматка);

3) провести пряму, перпендикулярно до продовження й другого шматка;

4) побудувати точку В – як середину між двома прямими;

5) побудувати пряму для лінеаризації заданої кусково-лінійної функції, що проходить через дві точки А і В, використовуючи стандартне рівняння прямої: $\dfrac{y - y_A}{x - x_A} = \dfrac{y_B - y_A}{x_B - x_A}$.

Для підстановки в рівняння прямої координати точок рівні $A\big(k, (k-1)\delta_1\big)$, $B\left(2k-1, \dfrac{(3k-3)\delta_1 + (k-1)\delta_2}{2}\right)$.

Тоді маємо $\dfrac{y-(k-1)\delta_1}{x-k}=\dfrac{(k-1)(\delta_1+\delta_2)}{k-1}$, після перетворення одержимо:

$$y-(k-1)\delta_1=(x-k)(\delta_1+\delta_2).$$

Частина d_1 в рівнянні (5.11) визначається таким способом:

$$d_1=\sigma_1\left(\frac{i+k-|i-k|}{2}-1\right)+\sigma_2\left(\frac{i-k+|i-k|}{2}\right)+d_1.$$

Отже, оптимізація дробово-лінійної функції з кусково-лінійною тенденцією знаменника полягає в розбитті функції знаменника (чисельника при умові кусково-лінійності) на частини і в подальшому дослідженні цих частин та зведенні до загального виду.

Приклад 5.5. Знайти відношення двох лінійних функцій – значення дробово-лінійної функції, коли коефіцієнти змінюються в чисельнику на +3, а у знаменнику спостерігається кусково-лінійна тенденція: перша частина коефіцієнтів збільшується на 2, друга – на 5. У таблиці 5.4 задані значення коефіцієнтів цільової функції, точки – вершини графа й значення цільової функції при $n=4$.

Необхідно визначити розташування верши – об’єктів конфігурації перестановки на підграфах загального графа при зміні значень кусково-лінійної функції.

Зазначимо, що для розв’язання задачі з кусково-лінійною тенденцією та з дробово-лінійною цільовою функцією є доцільним для представлення графа комбінаторної конфігурації використати новий метод генерування – метод переміщення максимального елементу. Розглянемо наступну таблицю.

Таблиця 5.4

Розкладання значень кусково-лінійної функції

c	2	5	8	11	14	*A*
d	3	5	7	11	15	

№	x_1	x_2	x_3	x_4	x_5	***ch***	***zn***	***chastka***	
1	1	2	3	4	5	150	153	0,98	
2	1	2	4	3	5	147	149	0,98	*max*

Продовд. табл. 5.4

№	x_1	x_2	x_3	x_4	x_5	*ch*	*zn*	*chastka*	
3	1	3	2	4	5	147	151	0,97	
4	1	3	4	2	5	141	143	0,98	
5	1	4	2	3	5	141	145	0,97	
6	1	4	3	2	5	138	141	0,97	
7	2	1	3	4	5	147	151	0,97	
8	2	1	4	3	5	144	147	0,97	
9	2	3	1	4	5	141	147	0,95	
10	2	3	4	1	5	132	135	0,97	
11	2	4	1	3	5	135	141	0,95	
12	2	4	3	1	5	129	133	0,96	
13	3	1	2	4	5	141	147	0,95	
14	3	1	4	2	5	135	139	0,97	
15	3	2	1	4	5	138	145	0,95	
16	3	2	4	1	5	129	133	0,96	
17	3	4	1	2	5	126	133	0,94	
18	3	4	2	1	5	123	129	0,95	
19	4	1	2	3	5	132	139	0,94	
20	4	1	3	2	5	129	135	0,95	
21	4	2	1	3	5	129	137	0,94	
22	4	2	3	1	5	123	129	0,95	
23	4	3	1	2	5	123	131	0,93	*min*
24	4	3	2	1	5	120	127	0,94	

В таблиці представлено розкладання значень кусково-лінійної функції на підграф вершин, для яких значення останньої координати дорівнює $x_5 = 5$.

На основі вищезазначених розрахунків побудуємо підграф, в якому вершини генеруються методом переміщення максимального елементу.

Даний граф є підграфом загального графа конфігурації перестановок $G(P_5)$, а вказана орієнтація ребер в графі, визначає зміну значень функції $F(x)$.

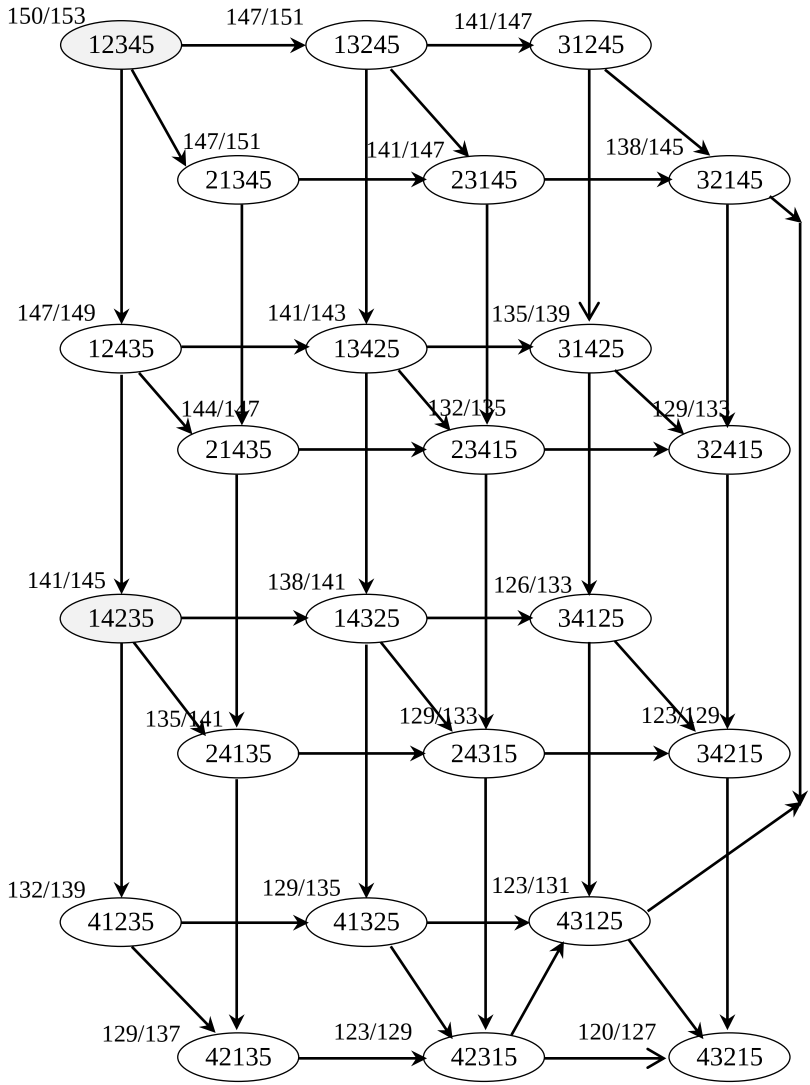


Рис. 5.7. Підграф *A* при максимальному значенні останнього елементу вершин

На основі вище розглянутого рис. 5.7 максимальне значення функції на даному підграфі досягається в точці (1 2 4 3 5) і рівне $\frac{147}{149}$, а мінімальне значення – в точці (4 3 1 2 5) і рівне $\frac{123}{131}$.

Таблиця 5.5

Розкладання значень цільової функції

c	2	5	8	11	14	*A*
d	3	5	7	11	15	

№	x_1	x_2	x_3	x_4	x_5	*ch*	*zn*	*chastka*	
1	1	2	3	5	4	147	149	0,99	
2	1	2	4	5	3	141	141	1,00	
3	1	3	2	5	4	144	147	0,98	
4	1	3	4	5	2	132	131	1,01	
5	1	4	2	5	3	135	137	0,99	
6	1	4	3	5	2	129	129	1,00	
7	2	1	3	5	4	144	147	0,98	
8	2	1	4	5	3	138	139	0,99	
9	2	3	1	5	4	138	143	0,97	
10	2	3	4	5	1	120	119	1,01	*max*
11	2	4	1	5	3	129	133	0,97	
12	2	4	3	5	1	117	117	1,00	
13	3	1	2	5	4	138	143	0,97	
14	3	1	4	5	2	126	127	0,99	
15	3	2	1	5	4	135	141	0,96	
16	3	2	4	5	1	117	117	1,00	
17	3	4	1	5	2	117	121	0,97	
18	3	4	2	5	1	111	113	0,98	
19	4	1	2	5	3	126	131	0,96	
20	4	1	3	5	2	120	123	0,98	
21	4	2	1	5	3	123	129	0,95	*min*
22	4	2	3	5	1	111	113	0,98	
23	4	3	1	5	2	114	119	0,96	
24	4	3	2	5	1	108	111	0,97	

Побудуємо наступний підграф загального графу в якому є фіксованим четвертий елемент.

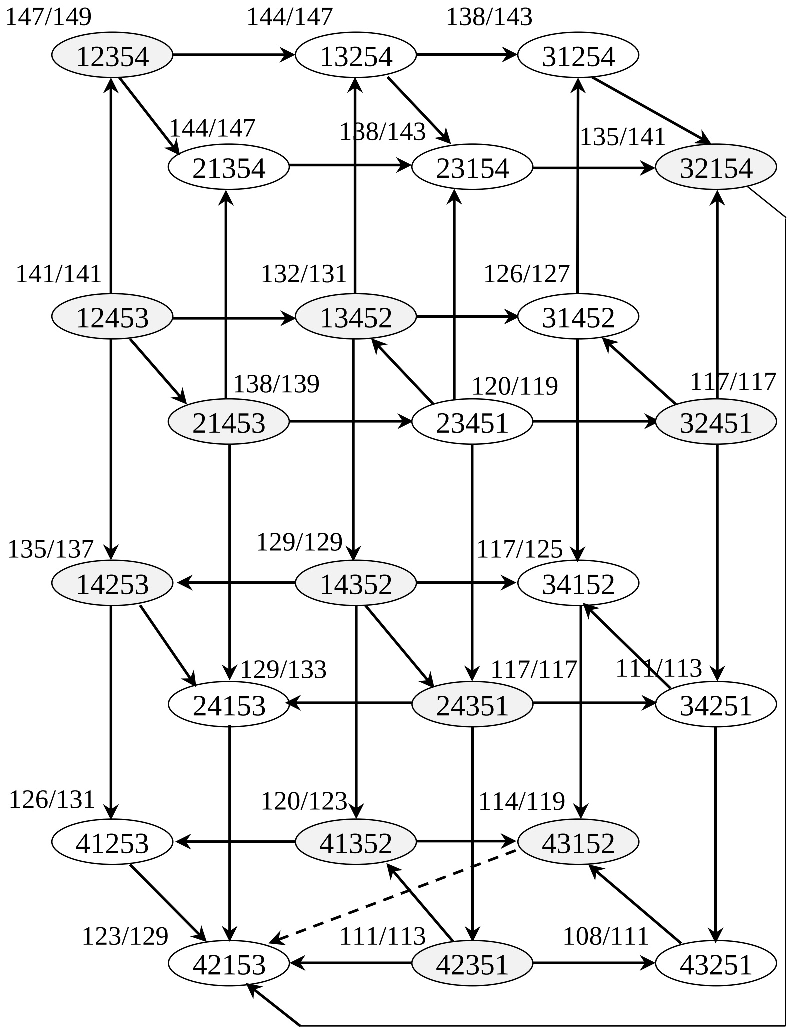


Рис. 5.8. Підграф *B* при фіксованому максимальному елементу на четвертому місці

Знову ж таки визначимо екстремальні значення функції на даному підграфі: максимальне значення функції рівне $\frac{120}{119}$ в точці (2 3 4 5 1), мінімальне – в точці (4 2 1 5 3), рівне $\frac{123}{129}$.

Таблиця 5.6

Розкладання значень цільової функції по підграфах

c	2	5	8	11	14	*A*
d	3	5	7	11	15	

№	x_1	x_2	x_3	x_4	x_5	*ch*	*zn*	*chastka*	
1	1	2	5	3	4	141	141	1,00	
2	1	2	5	4	3	138	137	1,01	
3	1	3	5	2	4	135	135	1,00	
4	1	3	5	4	2	129	127	1,02	
5	1	4	5	2	3	126	125	1,01	
6	1	4	5	3	2	123	121	1,02	
7	2	1	5	3	4	138	139	0,99	
8	2	1	5	4	3	135	135	1,00	
9	2	3	5	1	4	126	127	0,99	
10	2	3	5	4	1	117	115	1,02	
11	2	4	5	1	3	117	117	1,00	
12	2	4	5	3	1	111	109	1,02	*max*
13	3	1	5	2	4	129	131	0,98	
14	3	1	5	4	2	123	123	1,00	
15	3	2	5	1	4	123	125	0,98	
16	3	2	5	4	1	114	113	1,01	
17	3	4	5	1	2	105	105	1,00	
18	3	4	5	2	1	102	101	1,01	
19	4	1	5	2	3	117	119	0,98	
20	4	1	5	3	2	114	115	0,99	
21	4	2	5	1	3	111	113	0,98	*min*
22	4	2	5	3	1	105	105	1,00	
23	4	3	5	1	2	102	103	0,99	
24	4	3	5	2	1	99	99	1,00	

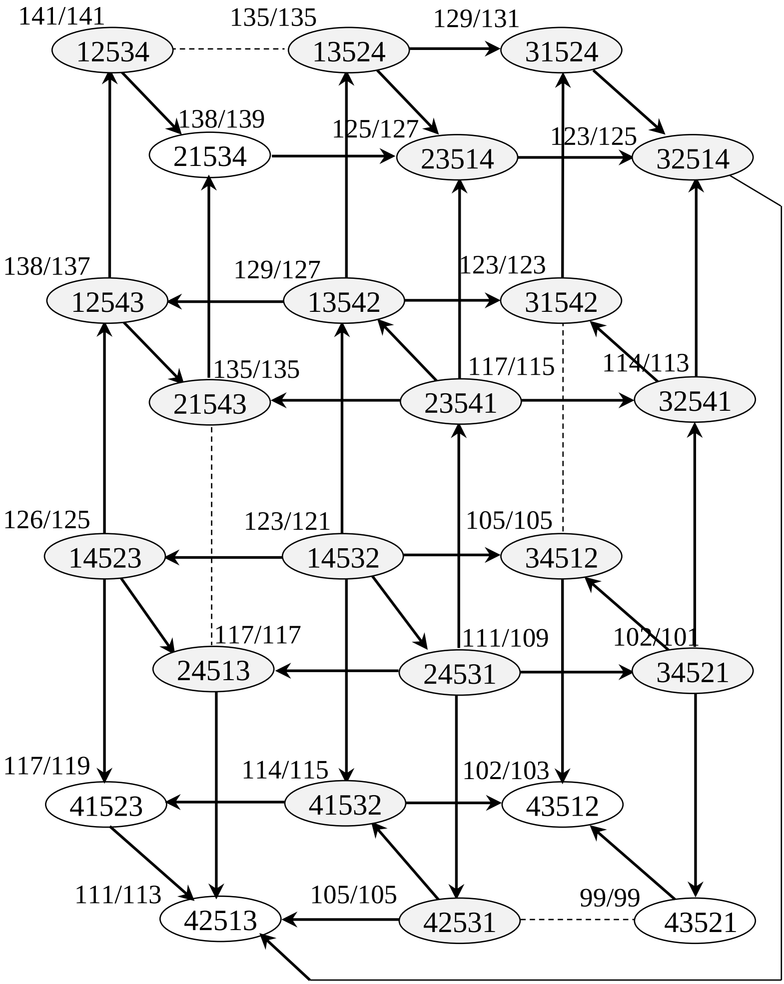


Рис. 5.9. Підграф *C* при фіксованому значенні максимального елементу $x_3 = 5$

В таблиці представлено розкладання значень кусково-лінійної функції на підграф вершин, для яких значення координати $x_3 = 5,$ тобто відбулося переміщення максимального елементу на третє місце.

На основі вище зазначених розрахунків побудуємо підграф, в якому вершини генеруються метод переміщення максимального елементу.

Аналізуючи зроблені розрахунки, та розміщення вершин на підграфі, визначаємо вершину, в якій значення функції мінімальне – до неї сходяться орієнтовані ребра (4 2 5 1 3), значення рівне $\frac{111}{113}$, максимальне значення $\frac{111}{109}$ досягається в вершині, від якої відходять орієнтовані ребра (2 4 5 3 1).

Розраховуємо наступну таблиці для перевірки, в якій розміщені вершини підграфа, для яких $x_2 = 5$.

Таблиця 5.7

Розкладання значень цільової функції

c	2	5	8	11	14	A
d	3	5	7	11	15	

№	x_1	x_2	x_3	x_4	x_5	*ch*	*zn*	*chastka*	
1	1	5	2	3	4	132	135	0,98	
2	1	5	2	4	3	129	131	0,98	
3	1	5	3	2	4	129	131	0,98	
4	1	5	3	4	2	123	123	1,00	
5	1	5	4	2	3	123	123	1,00	
6	1	5	4	3	2	120	119	1,01	
7	2	5	1	3	4	126	131	0,96	
8	2	5	1	4	3	123	127	0,97	
9	2	5	3	1	4	120	123	0,98	
10	2	5	3	4	1	111	111	1,00	
11	2	5	4	1	3	114	115	0,99	
12	2	5	4	3	1	108	107	1,01	max
13	3	5	1	2	4	117	123	0,95	
14	3	5	1	4	2	111	115	0,97	
15	3	5	2	1	4	114	119	0,96	
16	3	5	2	4	1	105	107	0,98	
17	3	5	4	1	2	102	103	0,99	
18	3	5	4	2	1	99	99	1,00	
19	4	5	1	2	3	105	111	0,95	min
20	4	5	1	3	2	102	107	0,95	
21	4	5	2	1	3	102	107	0,95	
22	4	5	2	3	1	96	99	0,97	
23	4	5	3	1	2	96	99	0,97	
24	4	5	3	2	1	93	95	0,98	
				max	1,01		min	0,95	

Будуємо за розрахунками відповідний підграф і визначаємо максимальне та мінімальне значення кусково-лінійної функції.

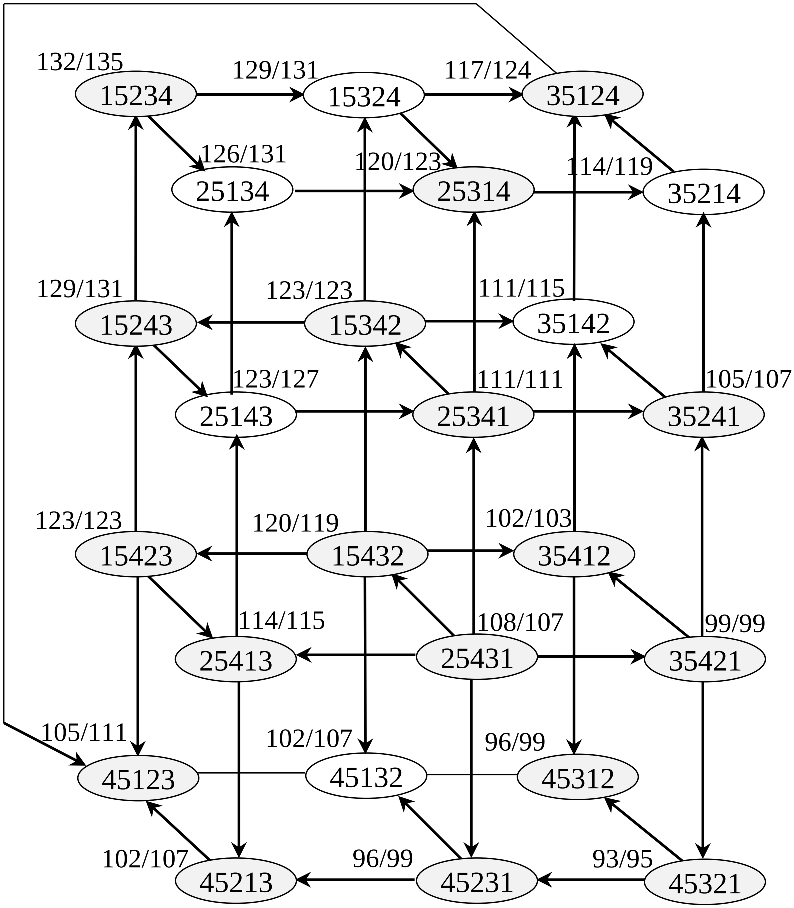


Рис. 5.10. Підграф D при фіксованому значенні максимального елементу на другому місці

Далі максимальний елемент переміщується на перше місце, тобто значення першої координати для вершин наступного підграфа рівне $x_1 = 5$.

Таблиця 5.8

Розкладання значень цільової функції по підграфах

c	2	5	8	11	14	A
d	3	5	7	11	15	

№	x_1	x_2	x_3	x_4	x_5	*ch*	*zn*	*chastka*	
1	5	1	2	3	4	120	127	0,94	
2	5	1	2	4	3	117	123	0,95	
3	5	1	3	2	4	117	123	0,95	
4	5	1	3	4	2	111	115	0,97	
5	5	1	4	2	3	111	115	0,97	
6	5	1	4	3	2	108	111	0,97	
7	5	2	1	3	4	117	125	0,94	
8	5	2	1	4	3	114	121	0,94	
9	5	2	3	1	4	111	117	0,95	
10	5	2	3	4	1	102	105	0,97	
11	5	2	4	1	3	105	109	0,96	
12	5	2	4	3	1	99	101	0,98	max
13	5	3	1	2	4	111	119	0,93	min
14	5	3	1	4	2	105	111	0,95	
15	5	3	2	1	4	108	115	0,94	
16	5	3	2	4	1	99	103	0,96	
17	5	3	4	1	2	96	99	0,97	
18	5	3	4	2	1	93	95	0,98	
19	5	4	1	2	3	102	109	0,94	
20	5	4	1	3	2	99	105	0,94	
21	5	4	2	1	3	99	105	0,94	
22	5	4	2	3	1	93	97	0,96	
23	5	4	3	1	2	93	97	0,96	
24	5	4	3	2	1	90	93	0,97	

Розрахунки в таблиці визначають значення функції в вершинах останнього підграфа, який розміщено в загальному графі нижче всіх.

Чисельний приклад наглядно характеризує важливість існування орієнтованого графа комбінаторної конфігурації перестановок для розв’язування екстремальних задач та його структуру.

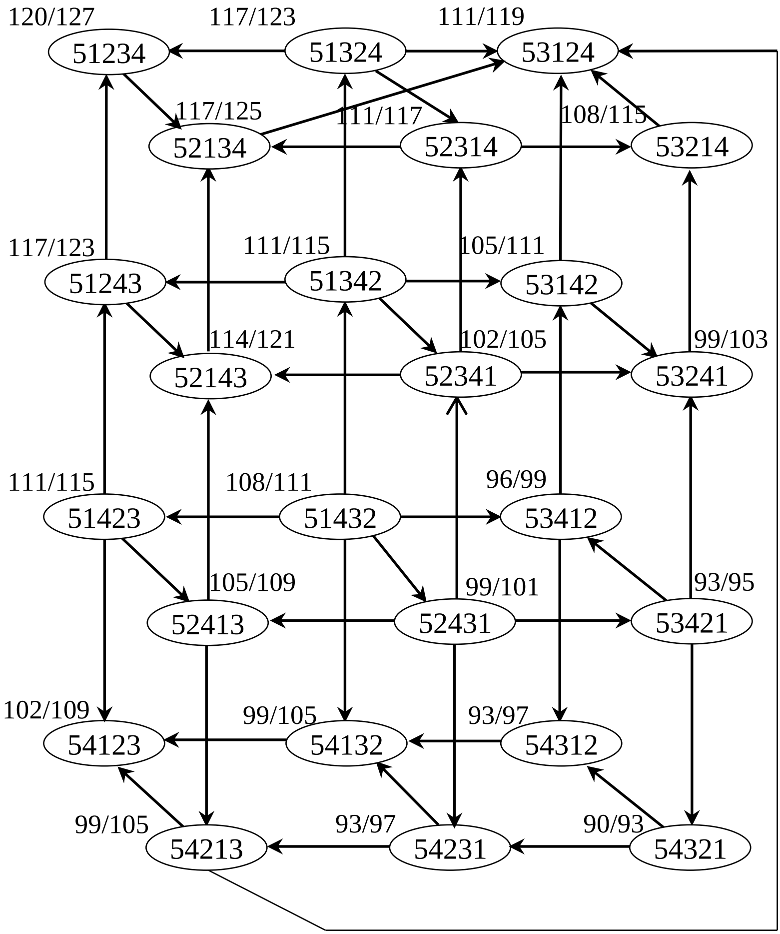


Рис. 5.11. Підграф E при фіксованому значенні максимального елементу на першому місці

Висновки до розділу

В розділі розглянута математична модель задачі з урахуванням комбінаторних властивостей області допустимих розв'язків та дробово-лінійних функцій критеріїв, яка може бути успішно застосована при розв'язанні різних практичних задач. Установлено взаємозв'язок між задачею на комбінаторній конфігурації перестановок і задачею на неперервній допустимій множині, а також обґрунтоване використання й застосування теорії графів для побудови методу розв'язування сформульованої задачі. На підставі доведених теорем, продовжуючи дослідження й розвиваючи результати попередніх робіт, запропонований підхід до розв'язання задачі з дробово-лінійними критеріями на допустимій множині комбінаторної конфігурації перестановок, що полягає у зведенні пошуку розв'язків вихідної задачі до задачі побудови гамільтонового шляху на графі конфігурації перестановок.

Зокрема, побудовано орієнтований граф конфігурації перестановок для дробово-лінійної функції і встановлено, що граф перестановок $\tilde{G}\left(P_n\right)$ для дробово-лінійної функції $F\left(x\right)$, коефіцієнти якої впорядковані згідно арифметичної прогресії, співпадає з графом перестановок для лінійної функції $G\left(P_n\right)$ з точністю до орієнтації.

Розглянуто більш складні екстремальні задачі з кусково-лінійною тенденцією знаменника та їх властивості. Зроблено числові експерименти в яких використовуються нові методи генерування комбінаторної конфігурації перестановок – рекурсивний метод та метод переміщення максимального елементу, проведено їх аналіз та дослідження.

РОЗДІЛ 6. ЗАГАЛЬНА СХЕМА РОЗВ'ЯЗУВАННЯ ЗАДАЧ НА КОМБІНАТОРНИХ КОНФІГУРАЦІЯХ

В даному розділі розглядається загальна схема методу направленого структурування для розв'язування екстремальних комбінаторних задач з додатковими умовами, що виникла на основі розглянутої в попередніх розділах задач оптимізації лінійної та дробово-лінійної функцій на комбінаторних конфігураціях без врахування додаткових умов, дослідження основних властивостей комбінаторних конфігурацій, а також встановленого графа генерації послідовності конфігурацій, за допомогою якого встановлюється зміна значень функцій.

Слід зазначити, що додаткові умови ускладнюють процес розв'язку задачі, але зменшують область допустимих розв'язків, звужуючи тим самим множину допустимих розв'язків.

6.1. Загальна постановка екстремальних задач на комбінаторних конфігураціях з додатковими обмеженнями та метод направленого структурування для їх розв'язання

Розглянемо екстремальну задачу комбінаторної оптимізації як обчислювальну проблему, у якій задана множина альтернатив $X=\{x\}$, цільова функція $f(x): X \to R$, і потрібно знайти альтернативу $x^0 \in X$, на якій ця цільова функція приймає екстремальне значення: $f(x^0)=\underset{x\in X}{extr}\, f(X)$, $extr \in \{min, max\}$. Для задач оптимізації альтернативи $x \in X$ звичайно називають допустимими розв'язками, x^0 – оптимальний розв'язок, $X=\{x\}$ – множиною допустимих розв'язків.

У даному випадку розглядається X – деяка задана комбінаторна конфігурація. Задача може містити також додаткові лінійні обмеження, які утворять опуклу многогранну множину $D \subset R^n$ вигляду: $D=\{x \in R^n \mid Gx \le h\}$, де $G \in R^{m\times n}$, $h \in R^m$. Запишемо лінійні обмеження у вигляді лінійних нерівностей:

$$G \cdot x = \sum_{j=1}^{n} g_{ij}\, x_j \ge h_i\, ;\, i \in N_m\, ;\, j \in N_n. \qquad (6.1)$$

Зазначимо, що в попередніх розділах розглядалася задача локалізації лінійної функції за заданим значенням. Є доцільним використати при розв'язанні даної задачі цей метод. Враховуючи, що задача локалізації розглядалася для екстремальної безумовної задачі з лінійною функцією, то для задачі з додатковими умовами даний метод необхідно застосувати для кожного додаткового обмеження і об'єднавши одержані результати, знайти загальний розв'язок.

Конкретизуємо задачу і розглянемо її на комбінаторній конфігурації перестановок. Розглянемо перестановки з множини $A=\{1,2,\dots,n\}$. Кількість елементів множини перестановки $P_n(A)$ дорівнює $n!$. послідовності перестановок, згідно з методом їх генерування з попередніх розділів інтерпретуються як граф G_n, вершини якого відповідають всім точкам множини перестановок $P_n(A)$. Для однокритеріальної задачі $F(x)=f(x)$ без додаткових обмежень максимальне значення лінійної функції $f(x)$ при упорядкуванні коефіцієнтів по зростанню на графі перестановок $G(P_n)$ досягається в перестановці $(1,2,\dots,n)$, а мінімальне – у перестановці $(n,n-1,\dots,2,1)$. Враховуючи умову (6.1), сформулюємо підзадачі, що необхідно розв'язати для визначення підмножини допустимих перестановок: визначити множину зв'язних пар перестановок $(\underline{x},\overline{x})$, для яких при заданому $h_i, i\in N_m$ має місце

$$\overline{x}=\underset{Gx\ge h_i}{arg\,min}\,G(x), \tag{6.2}$$

$$\underline{x}=\underset{Gx<h_i}{arg\,max}\,G(x). \tag{6.3}$$

Загальна схема даного алгоритму полягає в наступному:

Початковий етап: Початкова множина перестановок M_0 замінюється на базову M за допомогою вихідної перестановки u, яка нормалізує цільову функцію $f(x)$. За допомогою цієї ж перестановки переводимо множину перестановок кожної обмежуючої функції в базову (спільну для всіх функцій). Тепер для кожної i-ої обмежуючої функції g_i робимо такі операції:

1) нормалізуємо функцію за допомогою перестановки u_i, отримуючи індивідуальну множину перестановок M_i; тоді для

кожної обмежуючої функції граф перестановок $G(P_n)$ має стандартний вигляд:

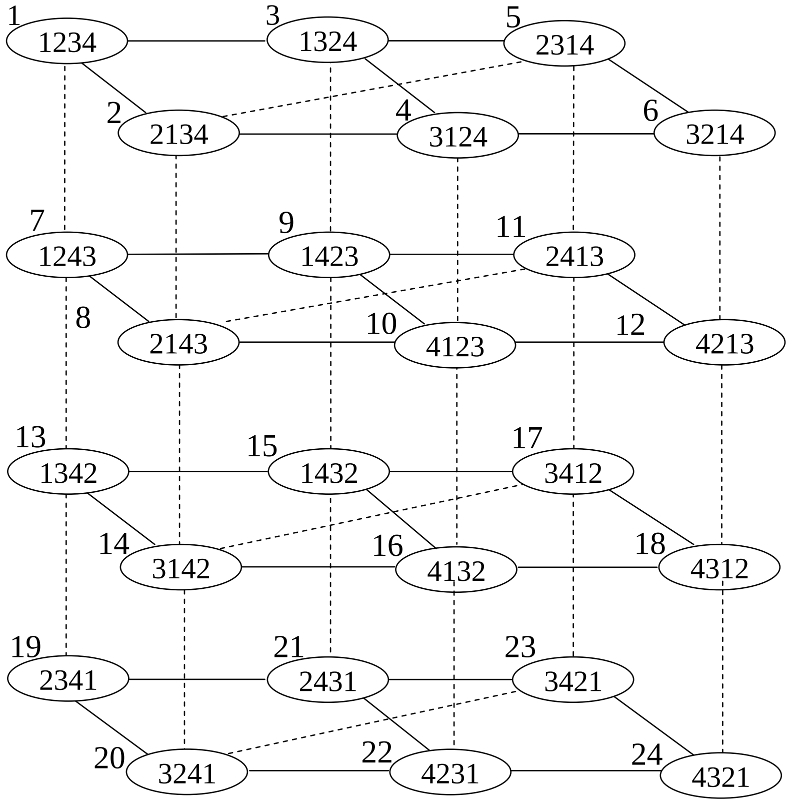


Рис. 6.1. Стандартний вигляд графа перестановок

2) розв'язуючи задачу локалізації для $h_i, i \in N_m$, отримуємо допустиму множину перестановок для g_i;

3) за допомогою перестановки u_i^{-1} визначаємо ту ж множину в базовій множині перестановок M.

4) знаходимо на цій множині *extr* функції g_i.

5) після всіх цих операцій відносно кожної обмежуючої функції знаходимо розв'язок задачі як $min\left\{extrg_j\right\}$.

6) за допомогою перестановки u^{-1} знаходимо розв’язок задачі в початковій множині перестановок M_0.

Перед тим, як розглянути приклад, детальніше розберемо в алгоритмі пункти 2 і 3. Підграф графа $G(P_n)$, у якого зафіксовано останній елемент i, назвемо гранню графа $G(P_n)$ і позначимо G_i. Всі грані будемо зображати у вигляді ребра, інцидентні вершини якого є початкова та кінцева вершини відповідного підграфа. В результаті отримаємо структурну схему графа $G(P_n)$, яка має вигляд драбини (рис. 6.2):

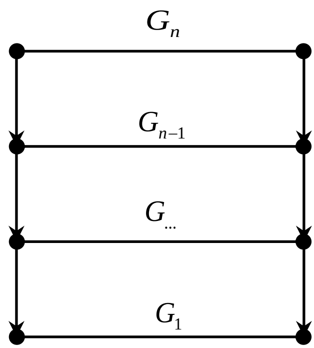


Рис. 6.2. Структурна схема графа $G(P_n)$

Аналогічно робиться структурна схема і підграфів G_i, тільки в них будуть фіксовані 2 індекси (і так далі по індукції). Виконуючи пункт 2 в алгоритмі, отримаємо спочатку структурну схему для обмежуючої функції $G_i(g_i)$, $(i = 1, 2, \dots, m)$. Подрібнюючи структурні схеми, отримаємо допустиму множину перестановок для g_i у вигляді підграфів з різною кількістю індексів. Тепер для виконання пункту 3 необхідно подіяти оберненою перестановкою u_i^{-1} на індекси цих підграфів і отримаємо шукану кількість таких же підграфів з індексами, але уже в базовій множині.

Розглянемо цей алгоритм на **прикладі 6.1:** дано функцію $f(x) = 4x_1 + 2x_2 + 7x_3 + 9x_4$, елементи множини перестановок для $P_4 = \{1, 2, 3, 4\}$; задані додаткові обмеження на перестановки, що визначають область допустимих значень цільової функції: $g_1 = 5x_1 + 6x_2 + 10x_3 + 1x_4 \le 60$, $g_2 = 4x_1 + 6x_2 + 8x_3 + 9x_4 \ge 63$.

Знайти: екстремум цільової функції – максимум.

Розв'язання: **Початковий етап.** Нормалізуємо задану цільову функцію $f(x)$ у порядку зростання за допомогою перестановки $u = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$. Отримаємо цільову функцію $f(x) = 2x_1 + 4x_2 + 7x_3 + 9x_4$. Граф перестановок для нормалізованих функцій має вигляд рис. 6.1. Нормалізуємо також обмежуючі функції $g_1 = 6x_1 + 5x_2 + 10x_3 + 1x_4$ та $g_2 = 6x_1 + 4x_2 + 8x_3 + 9x_4$ за допомогою перестановки, що і цільову функцію $u = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$.

Крок 1. Нормалізуємо g_1 за допомогою перестановки $u_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$ до вигляду $g_1 = x_1 + 5x_2 + 6x_3 + 10x_4$.

Структурна схема графу G розбивається на структурні схеми підграфів G_1, G_2, G_3, G_4, і знаходиться значення функції в крайніх точках підграфів, що визначають *min* та max (рис. 6.3):

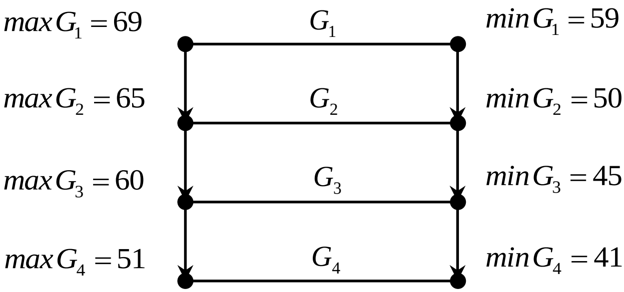


Рис. 6.3. Загальна структурна схема графу для g_1

Розв'язуючи далі задачу про локалізацію, отримаємо в G_1 дві вершини, які не задовольняють обмеженням, це $4=(3124)$ та $6=(3214)$. В G_2 це такі вершини 10, 11, 12, в G_3 всі, крім 13, а в G_4 всі вершини задовольняють обмеженням. В сумі маємо множину вершин (4, 6, 10, 11, 12, 13, 19, 20, 21, 22, 23, 24). За допомогою оберненої перестановки $u_1^{-1}=\begin{pmatrix}1 & 2 & 3 & 4\\ 4 & 2 & 1 & 3\end{pmatrix}$ з цієї множини в базовій множині отримаємо такі перестановки (8, 7, 2, 15, 1, 24, 18, 12, 17, 6, 11, 5), які не задовольняють обмеженням. Без сумніву, максимальне значення цільова функція приймає у точці $3=(1324)$, яке дорівнює 64. Зауважимо, що підграфу G_4, вершини якого задовольняють умовам функції g_1, відповідає в базовій множині підграф G_3, тому що в підстановці u_i^{-1} 4 переходить в 3.

Крок 2. Нормалізуємо g_2 за допомогою перестановки $u_2=\begin{pmatrix}1 & 2 & 3 & 4\\ 2 & 1 & 3 & 4\end{pmatrix}$ до вигляду $g_2=4x_1+6x_2+8x_3+9x_4$.

Структурна схема графу G розбивається на структурні схеми підграфів G_1, G_2, G_3, G_4, і визначається значення функції в крайніх точках підграфів, які визначають мінімальне та максимальне значення на підграфах

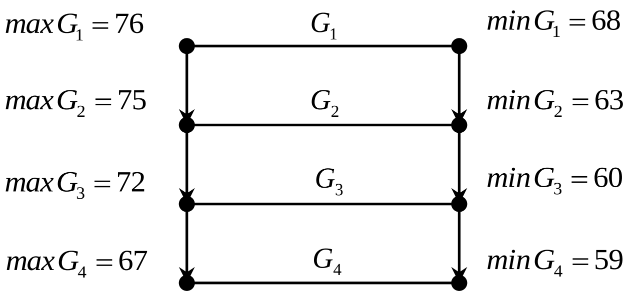


Рис. 6.4. Загальна структурна схема графу для g_2

Всі вершини підграфів G_1 та G_2, що зображені на структурній схемі графа 6.4, задовольняють обмеженням. В G_3 їм не задовольняють дві вершини 17 та 18. В G_4 обмеженням не задовольняють три вершини 22, 23 та 24. За допомогою оберненої перестановки $u_2^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$ з цієї множини в базовій множині отримаємо такі самі перестановки. Оскільки вершина 2 входить в обмеження і для функції g_2, то розв'язок задачі не зміниться – максимальне значення цільова функція приймає у точці $3 = (1324)$, яке дорівнює 64.

Залишається повернутися у початкову множину перестановок: максимальний розв'язок задачі буде на перестановці (3124).

6.2. Застосування методу направленого структурування до розв'язуваня екстремальних задач на комбінаторних конфігураціях розміщень

Враховуючи те, що для конфігурації перестановок існує граф, який може бути застосований в методі направленого структурування, то згідно з властивостями конфігурації розміщень, можна також побудувати граф розміщень.

Розглянемо граф конфігурації розміщень $A(n,m)$, яка утворена з елементів $\{1,2,\dots,n\}$ при умові $A(4,3)$. Тоді число таких конфігурацій дорівнює

$$A(n,m) = n(n-1)\dots(n-m+1) = (n)_m,\ m \le n,$$

де $(n)_m = \dfrac{n!}{(n-m)!}$ і $A(n,m) = 0, m > n,$ а для $A(4,3) = 24.$

На основі вище зазначених роздумів та згідно нового методу генерування комбінаторних об'єктів – методу переміщення максимального елементу можна сформулювати наступну теорему.

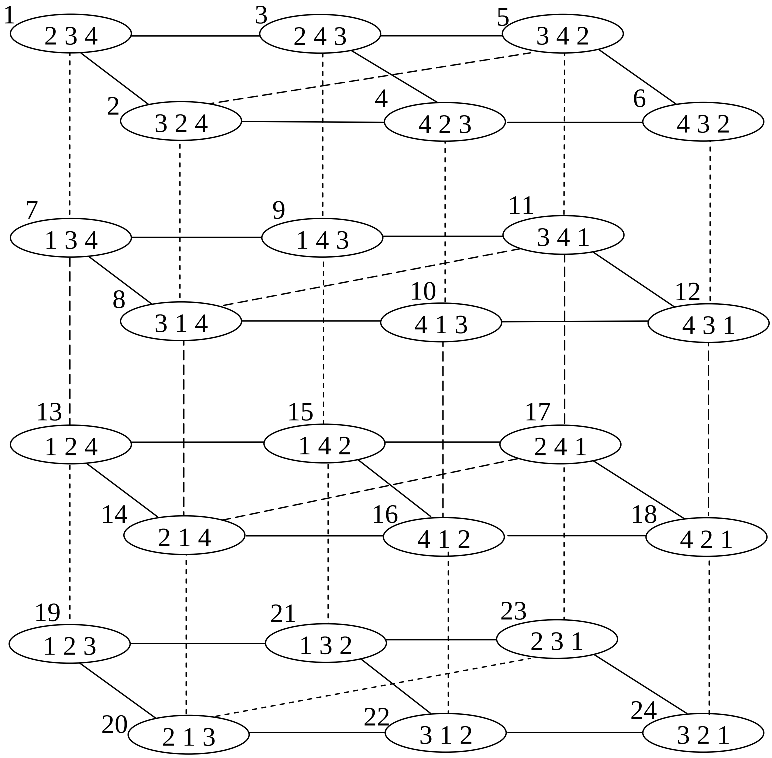


Рис. 6.5. Граф конфігурації розміщень

Теорема 6.1. Граф $G\left(A(n,m)\right)$ конфігурації розміщень $A(n,m)$ при $n > m$ і при будь-якому векторі $\vec{A} = \{a_1, a_2, \ldots, a_n\}$ еквівалентний графу $G\left(P_n\right)$ комбінаторної конфігурації перестановок P_n.

Доведення теореми випливає з означення еквівалентності графів та згідно методу генерування – методу максимального переміщення елементів. Тоді метод направленого структурування може бути без обмежень застосований і до екстремальних задач з додатковими обмеженнями на розміщеннях.

Загальна схема алгоритму за методом направленого структурування до комбінаторної конфігурації розміщень полягає в наступному:

Початковий етап. Початкова множина розміщень A_0 заміняється на базову A за допомогою вихідної перестановки u, яка нормалізує цільову функцію $f(x)$. За допомогою цієї ж перестановки переводимо множину перестановок кожної обмежуючої функції в базову (спільну для всіх функцій). Тепер для кожної i-ої обмежуючої функції g_i робимо такі операції:

1) нормалізуємо функцію за допомогою перестановки

$$u_{\mathrm{i}} = u_i = \begin{pmatrix} 1 & 2 & \ldots & m \\ 2 & 1 & \ldots & \varphi_m \end{pmatrix}, \tag{6.4}$$

отримуючи індивідуальну множину розміщень A_i; тоді для кожної обмежуючої функції граф розміщень $G\big(A(n,m)\big)$ має стандартний вигляд

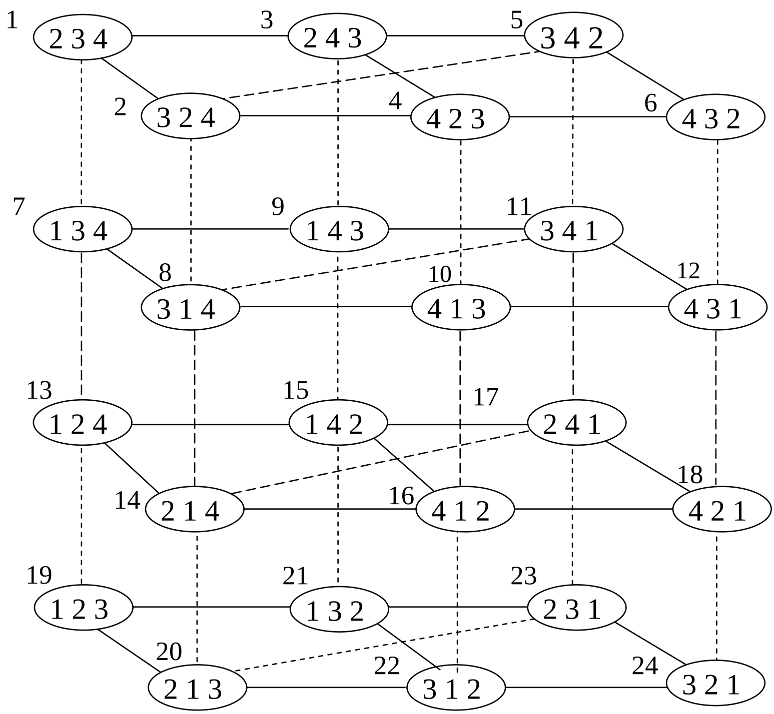


Рис. 6.6. Стандартний вигляд графа конфігурації розміщень

2) розв’язуючи задачу локалізації для $h_i, i \in N_m$, отримуємо допустиму множину розміщень для додаткового лінійного обмеження g_i;

3) за допомогою перестановки u_i^{-1} визначаємо ту ж множину в базовій множині розміщень $G\left(A\left(n,m\right)\right)$.

4) знаходимо на цій множині *extr* функції g_i.

5) після всіх цих операцій відносно кожної обмежуючої функції знаходимо розв’язок задачі як $min\left\{extrg_j\right\}$.

6) за допомогою перестановки $u^{-1}u^{-1}$ знаходимо розв’язок задачі в початковій множині розміщень A_0.

Розглянемо цей алгоритм на прикладі.

Приклад 6.3. Дано функцію

$$f\left(x\right) = 4x_1 + 2x_2 + 7x_3,$$

елементи множини, що формують множину розміщень $A_i\left(1234\right)$*;* задані додаткові обмеження, які визначають область допустимих значень цільової функції:

$$g_1 = 4x_1 + 6x_2 + 8x_3 \geq 42,\ \ g_2 = 5x_1 + 6x_2 + 10x_3 \leq 64,$$

Знайти: екстремум цільової функції – максимум.

Розв’язання:

Початковий етап. Нормалізуємо задану цільову функцію $f\left(x\right)$ у порядку зростання за допомогою перестановки

$$u = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

отримаємо цільову функцію: $f\left(x\right) = 2x_1 + 4x_2 + 7x_3$, а також обмежуючі функції

$$g_1 = 6x_1 + 4x_2 + 8x_3,$$

$$g_2 = 6x_1 + 5x_2 + 10x_3.$$

Граф розміщень для нормалізованих функцій має вигляд:

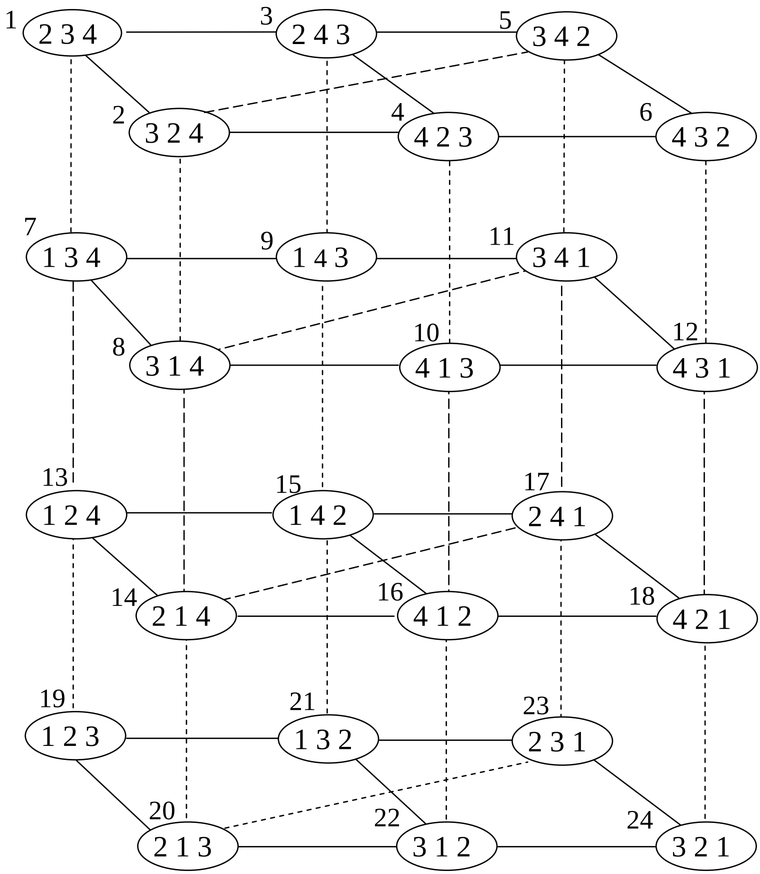


Рис. 6.7. Граф конфігурації розміщень

Крок 1. Нормалізуємо g_1 за допомогою перестановки $u_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ до вигляду $g_1 = 4x_1 + 6x_2 + 8x_3$.

Загальна структурна схема графу G як і в попередніх випадках, розбивається на структурні схеми підграфів G_1, G_2, G_3, G_4, і знаходиться значення функції в крайніх точках підграфів, які визначають мінімальне та максимальне значення на підграфах (рис. 6.8);

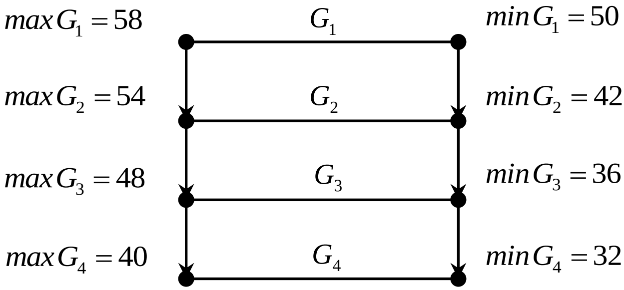


Рис. 6.8. Загальна структурна схема графу для g_1

Розв'язуючи далі задачу про локалізацію та враховуючи, що для перестановок, які можуть бути розв'язками задачі, необхідно виконання умови $g_1 = 4x_1 + 6x_2 + 8x_3 \geq 42$, згідно рис. 6.8. розв'язками будуть всі вершини підграфів G_1, G_2. А з множини вершин підграфа G_3 необхідно розглянути наступні:

№ п/п вершини	x_1	x_2	x_3	g_1
13	1	2	4	48
14	1	4	2	44
15	2	1	4	46
16	2	4	1	40
17	4	1	2	38
18	4	2	1	36

Тоді з G_3 отримаємо вершини, які задовольняють обмеженням, це $13 = (124)$, $14 = (142)$, $15 = (214)$. В G_4 всі вершини не задовольняють обмеженням. В сумі маємо множину вершин (1–6, 7–12, 13, 14, 15). За допомогою оберненої перестановки $u_1^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ з цієї множини в базовій множині отримаємо такі підграфи G_1, G_2 та точки перестановки (13, 14, 15), які задовольняють обмеженням. Максимальне значення цільова функція, при умові виконання першого додаткового обмеження

$g_1 = 4x_1 + 6x_2 + 8x_3 \geq 42$, приймає у точці $2 = (3\ 2\ 4)$, яке дорівнює 44.

Крок 2. Нормалізуємо g_2 за допомогою перестановки $u_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ до вигляду $g_2 = 5x_1 + 6x_2 + 10x_3 \leq 64$.

Схема структурного графу G розбивається на схеми структурних підграфів G_1, G_2, G_3, G_4, і визначається значення функції в крайніх точках підграфів, які визначають мінімальне та максимальне значення на підграфах

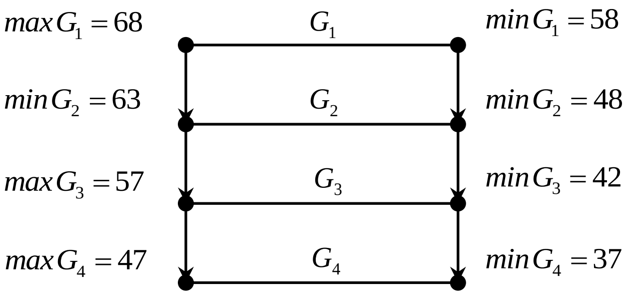


Рис. 6.9. Структурна схема графу для g_2

Всі вершини під графів G_1, G_2 та G_3 задовольняють обмеженням. В G_4 обмеженням задовольняють наступні вершини т. 3 $(2\ 4\ 3)$, т. 4 $(3\ 4\ 2)$, т. 5 $(4\ 2\ 3)$, т. 6 $(4\ 3\ 2)$.

За допомогою оберненої перестановки $u_2^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ з цієї множини в базовій множині отримаємо такі самі перестановки.

Оскільки вершина 2 в якій досягається максимальне значення цільової функції при виконанні першого додаткового обмеження g_1 не задовольняє додаткове обмеження g_2, то розв’язок задачі зміниться – максимальне значення цільова функція приймає у точці $8 = (3\ 1\ 4)$, яке дорівнює 42.

Повернемося у початкову множину розміщень за допомогою оберненого перетворення: максимальне значення функції $f(x)$ рівне 42 і досягається в точці 8 з координатами (314).

6.3. Застосування методу направленого структурування до розв'язування екстремальних задач на конфігурації сполучень

Розглянемо більш простий випадок застосування методу направленого структурування до розв'язування екстремальних задач на комбінаторній конфігурації сполучень.

Для конфігурації сполучень елементи можна зобразити в вигляді дерева, оскільки елементи сполучень згідно методу генерування можна розмістити в вигляді неорієнтованого графу, що не має циклів, тобто такого, в якому кожна пара вершин з'єднана одним простим шляхом (ланцюгом).

Дерево можна орієнтувати, вибравши для цього довільну вершину в якості кореня і ребрам приписати таку орієнтацію, що б кожна вершина з'єднувалася з коренем тільки одним простим шляхом, тоді елементи конфігурації сполучень будуть розміщені в вигляді орієнтованого дерева, де піддерево визначається початковим елементом в вершині.

Наприклад, нехай кількість усіх сполучень без повторень з n елементів по r рівне C_n^r, де r і n – невід'ємні цілі числа, причому $r \le n$, то базовим буде орієнтоване дерево з n піддеревами, на які можна розбити базове.

Слід зазначити, що при розв'язуванні екстремальних задач на множині сполучень не має значення порядок розміщення елементів, а лише їх вибір з вихідної множини $A = \left(a_1, a_2, \ldots, a_n\right)$, в якій елементи впорядковані за зростанням $a_1 \le a_2 \le \ldots \le a_n$, то екстремальне значення функції завжди можна визначити. Причому, між вершинами в кожному з піддерев існує гамільтонів шлях. Досить цікавими є задачі на множині сполучень, в яких необхідно оперувати з симетричними функціями типу

$$\alpha x_1 x_2 + \alpha x_2 x_3 + \ldots + \alpha x_{n-1} x_n . \qquad (6.5)$$

Оскільки такі функції є симетричними, то немає потреби в нормалізації функції. Тоді загальна схема алгоритму за методом направленого структурування для екстреамльних задач на сполученнях полягає в наступному:

Початковий етап. Вибирається базова M множина за допомогою вихідної перестановки u. Тепер для кожної i-ї обмежуючої функції g_i робимо такі операції:

1) визначаємо для кожної обмежуючої функції дерево $G(P_n)$, що має стандартний вигляд отримуючи індивідуальну множину сполучень M_i;

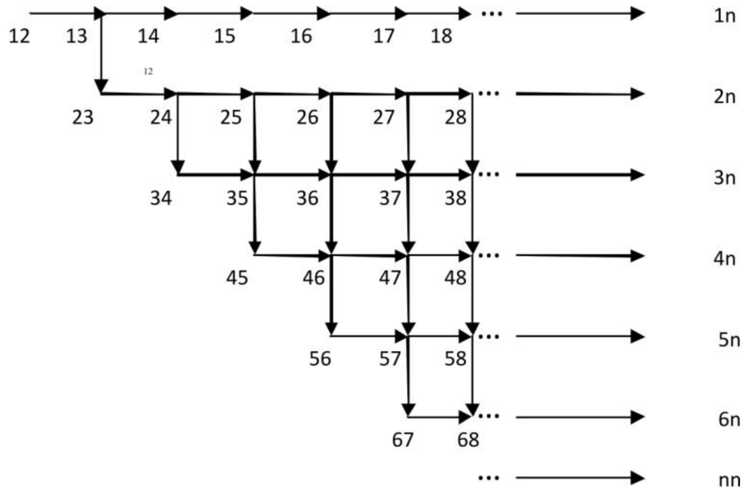


Рис. 6.10. Стандартний вигляд дерева сполучень

2) розв'язуючи задачу локалізації для $h_i, i \in N_m$, отримуємо допустиму множину сполучень для g_i;

3) за допомогою перестановки u_i^{-1} визначаємо ту ж множину в базовій множині сполучень M.

4) знаходимо на цій множині *extr* функції g_i.

5) після всіх цих операцій відносно кожної обмежуючої функції знаходимо розв'язок задачі як $min\{extrg_j\}$.

6) за допомогою перестановки u^{-1} знаходимо розв'язок задачі в початковій множині сполучень M_0.

Розглянемо цей алгоритм на прикладі.

Приклад 6.4. Дано функцію $f(x) = 6x_1x_2 + 6x_1x_3 + 6x_2x_3$, елементи множини, що формують множину сполучень $A_i(1\ 2\ 3\ 4\ 5\ 6)$; задані додаткові обмеження, які визначають область допустимих значень цільової функції:

$$g_1 = 4x_1x_2 + 4x_2x_3 + 4x_1x_3 \geq 165,\ \ g_2 = 5x_1x_2 + 5x_2x_3 + 5x_1x_3 \leq 220\ .$$

Знайти: екстремум цільової функції – максимум.

Розв’язання:

Початковий етап. Нормалізуємо задану цільову функцію $f\left(x\right)$ у порядку зростання за перестановкою $u=\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$; отримаємо цільову функцію $f\left(x\right)=6x_1x_2+6x_1x_3+6x_2x_3$, а також обмежуючі функції $g_1=4x_1x_2+4x_2x_3+4x_1x_3$, $g_2=5x_1x_2+5x_2x_3+5x_1x_3$.

Розглянемо конфігурацію сполучень з 6 по 3, де $C_6^3=20$.

Формуємо базове дерево елементів сполучення для цільової функції: вибираємо з впорядкованої за зростанням множини з шести заданих елементів 1, 2, 3, 4, 5, 6 перші три 1, 2, 3 і для побудови першого піддерева останній елемент замінюємо наступними з заданої множини, маємо: 1, 2, 4; 1, 2, 5; 1, 2, 6. Тоді друге піддерево утворюється шляхом заміни в кожному з попередніх елементів крім першого другої компоненти на наступний елемент з заданої множини, маємо 1, 3, 4; 1, 3, 5; 1, 3, 6.

Маємо базове дерево

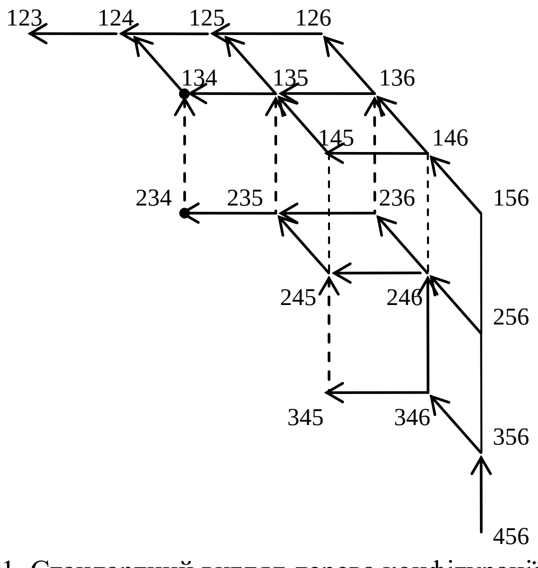


Рис. 6.11. Стандартний вигляд дерева конфігурації сполучень

На основі базового дерева будуємо схему структурного дерево рішень.

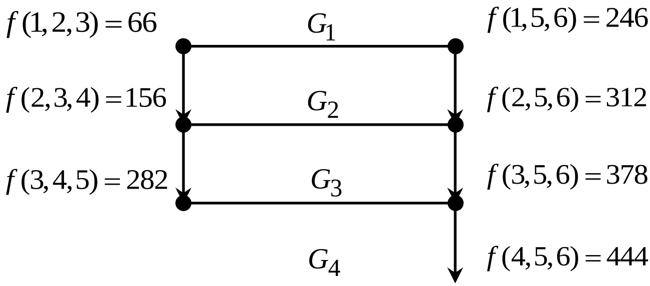


Рис. 6.12. Структурна схема піддерева сполучень для функції цілі $f(x)$

Крок 1. Нормалізуємо g_1 за допомогою перестановки $u_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ до вигляду $g_1 = 4x_1x_2 + 4x_2x_3 + 4x_1x_3$.

Розглядаємо структурну схему дерева конфігурації сполучень, що можна представити наступним чином

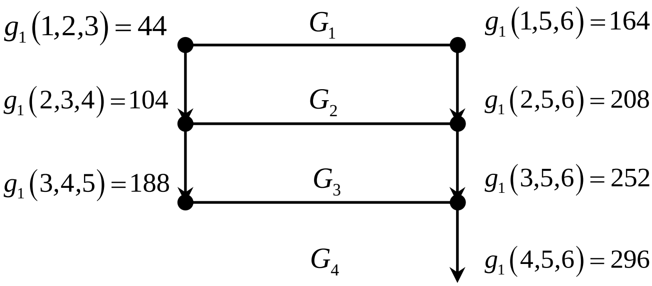


Рис. 6.13. Структурна схема піддерева сполучень для g_1

Структурна схема дерева G розбивається на піддерева G_1, G_2, G_3 і визначається значення функції в крайніх точках піддерев, які визначають мінімальне та максимальне значення на піддеревах (рис. 6.2).

Розв'язуючи далі задачу про локалізацію та враховуючи, що для сполучень, які можуть бути розв'язками задачі, необхідно виконання умови $g_1 = 4x_1x_2 + 4x_2x_3 + 4x_1x_3 \geq 165$, згідно рис. 6.13 розв'язками будуть всі вершини піддерев G_3, G_4. А з множини вершин піддерев G_1 – не задовольняє жодна з вершин, G_2 необхідно розглянути наступні (2 4 6), (2 5 6).

Крок 2. Нормалізуємо g_2 за допомогою перестановки $u_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ до вигляду $g_2 = 5x_1x_2 + 5x_2x_3 + 5x_1x_3$. Структурна схема дерева розбивається на піддерева G_1, G_2, G_3, G_4, для яких визначається значення функції в крайніх вершинах, що задовольняють обмеження $g_2 = 5x_1x_2 + 5x_2x_3 + 5x_1x_3 \leq 220$.

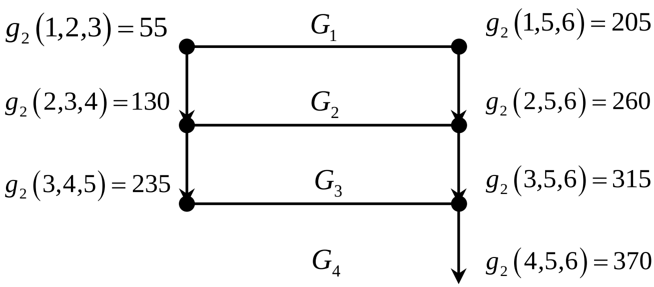


Рис. 6.14. Структурна схема дерева сполучень для g_2

Всі вершини піддерева G_1 задовольняють обмеженням. В G_2 обмеженням задовольняють лише наступні точки (2 3 4), (2 3 5), (2 3 6), (2 4 5), (2 4 6). В G_3 і G_4 обмеженням не задовольняють всі вершини.

Шукаємо перетин структурних схем піддерев для обох додаткових обмежень – це вершини піддерева G_2 .

Одержимо вершину (2 4 6), що задовольняє обидва обмеження, а функція набуває максимального значення в цій вершині, то точка (2 4 6) є розв'язком задачі.

6.4. Застосування методу направленого структурування до розв’язування екстремальних задач з дробово-лінійними цільовими функціями

Розглянемо дробово-лінійну функцію $F(x)$, де чисельник $f(x)$ і знаменник $g(x)$ – дві лінійні, в яких коефіцієнти визначаються за правилом арифметичної прогресії:

$$c_i = c_1 + \Delta(i-1),$$

$$d_i = d_1 + \delta(i-1). \tag{6.6}$$

Як відомо з розділу 4 можна побудувати гамільтонів шлях усередині кожної шестірки, елементів перестановки і прослідити зміни значень цільової функції у вершинах графа, так як і для лінійної функції. Тоді граф перестановок $\tilde{G}(P_n)$, згідно теореми 4.4. для дробово-лінійної функції $F(x)$, коефіцієнти якої визначені згідно (6.6), співпадає з графом перестановок для лінійної функції $G(P_n)$ з точністю до орієнтації. Тому метод направленого структурування можна застосувати до розв’язування задач з дробово-лінійними функціями цілі та з лінійними додатковими обмеженнями.

Для задач такого типу загальна схема даного алгоритму полягає в наступному:

Початковий етап. Визначаються коефіцієнти цільової функції за формулою (6.6) для чисельника і знаменника відповідно. Задається множна перестановок (якщо задача розглядається на множині перестановок).

Початкова множина перестановок M_0 замінюється на базову M за допомогою вихідної перестановки u, яка нормалізує цільову функцію $F(x)$. За допомогою цієї ж перестановки переводимо множину перестановок кожної обмежуючої функції в базову (спільну для всіх функцій). Тепер для кожної i-ї обмежуючої функції g_i робимо такі операції:

1) нормалізуємо функцію за допомогою перестановки u_i, отримуючи індивідуальну множину перестановок M_i; тоді для

кожної обмежуючої функції граф перестановок $G(P_n)$ має стандартний вигляд згідно рис. 6.1;

2) розв'язуючи задачу локалізації для $h_i, i \in N_m$, отримуємо допустиму множину перестановок для g_i;

3) за допомогою перестановки u_i^{-1} визначаємо ту ж множину в базовій множині перестановок *M;*

4) знаходимо на цій множині *extr* функції g_i*;*

5) після всіх цих операцій відносно кожної обмежуючої функції знаходимо розв'язок задачі як $min\{extrg_j\}$*;*

6) визначаємо максимальне значення дробово-лінійної функції $F(x)$ на об'єднаній множині вершин, що задовольняє кожне додаткове лінійне обмеження g_j;

7) за допомогою перестановки u^{-1} знаходимо розв'язок задачі з дробово-лінійною цільовою функцією в початковій множині перестановок M_0.

Розглянемо на прикладі реалізацію даного алгоритму.

Приклад 6.6. Дано функцію $F(x) = \dfrac{f(x)}{g(x)}$, в якій чисельник і знаменник виражається наступним чином

$$f(x) = 2x_1 + 5x_2 + 8x_3 + 11x_4 + 14x_5,$$

$$g(x) = 3x_1 + 5x_2 + 7x_3 + 11x_4 + 15x_5$$

задано елементи множини $A_i(1\ 2\ 3\ 45)$, що формують множину перестановок; додаткові обмеження, які визначають допустимі значення цільової функції: $g_1 = 4x_1 + 6x_2 + 8x_3 - 3x_4 - 1x_5 \geq 66$.

Знайти: екстремум цільової функції – максимум.

Розв'язання:

Початковий етап. Розглядаємо додаткове обмеження-функції g_1. Нормалізуємо задану функцію g_1 у порядку зростання за допомогою перестановки $u = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 4\,5\,1\,2\,3 \end{pmatrix}$ отримаємо обмежуючу функцію $g_1 = 4x_1 + 6x_2 + 8x_3 - 3x_4 - 1x_5 \geq 66$. На основі розрахунків четвертого розділу представимо граф перестановок для нормалізованих дробово-лінійних функцій вершини якого генеруються метод переміщення максимального елемента.

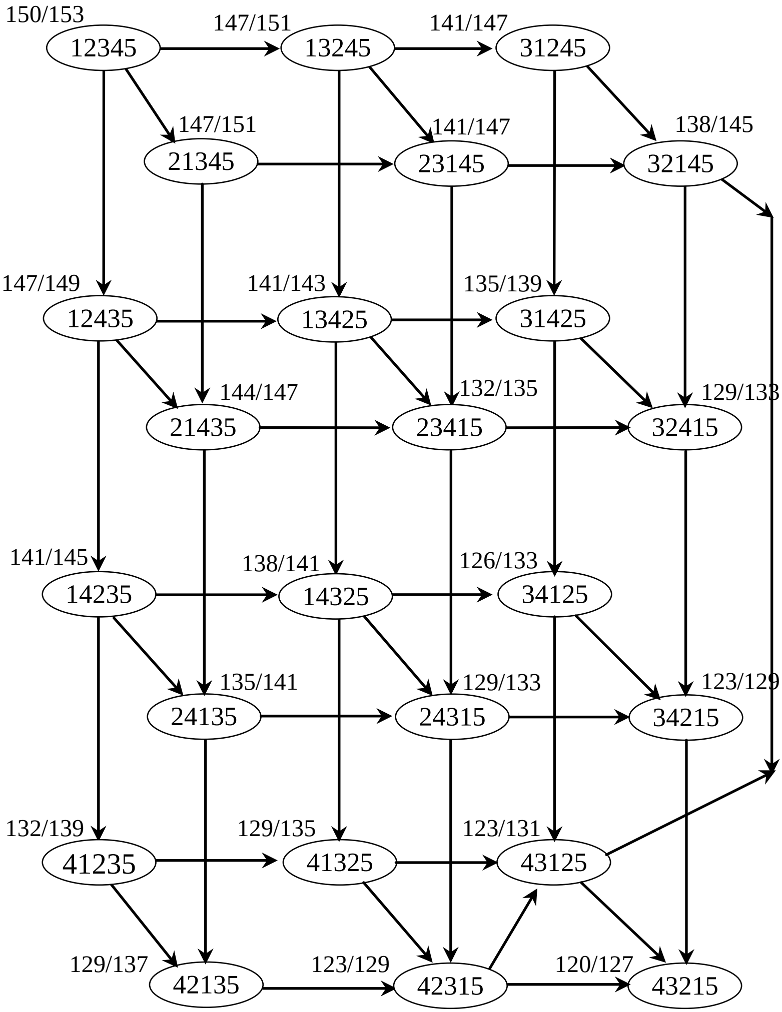


Рис. 6.15 Підграф *А* при фіксованому значенні останнього максимального елемента

Побудуємо наступний підграф загального графу в якому є фіксованим четвертий елемент, тобто граф отримується з попереднього шляхом переміщення максимального елемента з п’ятого на четверте місце.

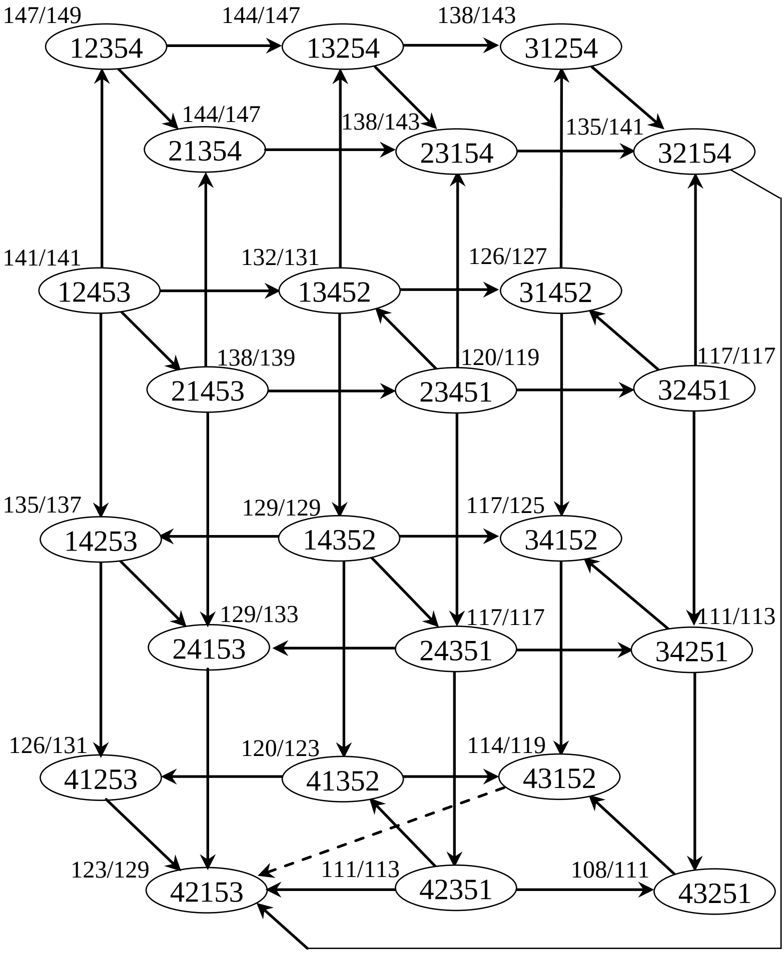


Рис. 6.16. Підграф *B* при фіксованому максимальному елементі на четвертому місці

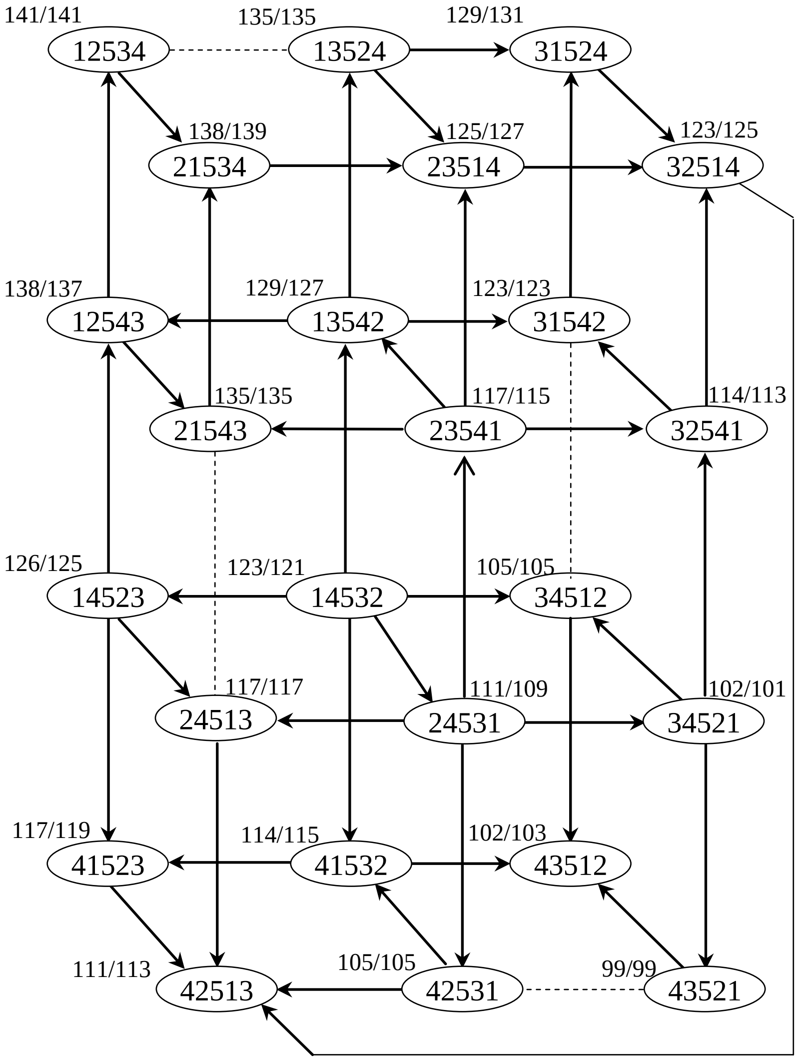


Рис. 6.17. Підграф C при фіксованому максимальному елементі на третьому місці

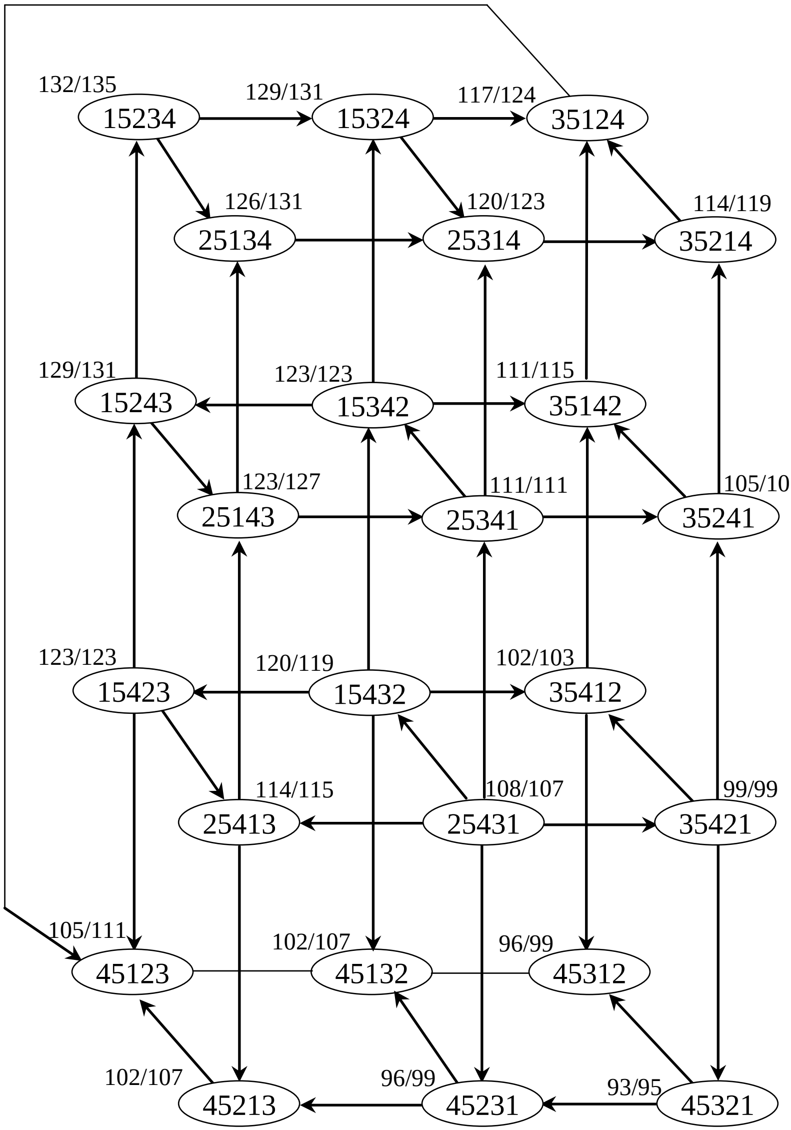


Рис. 6.18. Підграф *D* при фіксованому максимальному елементі на другому місці

Далі максимальний елемент переміщується на останнє місце.

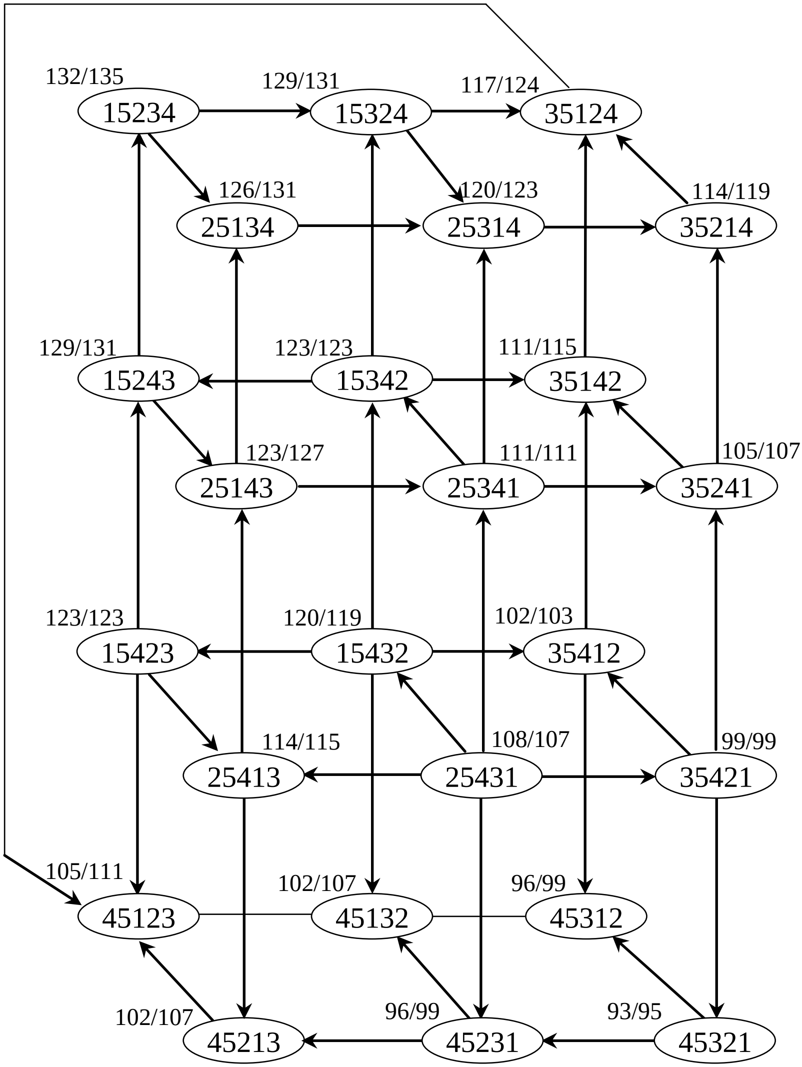


Рис. 6.19 Підграф *E* при фіксованому максимальному елементі на першому місці

Дане додаткове обмеження $g_1 = 4x_1 + 6x_2 + 8x_3 - 3x_4 - 1x_5 \geq 66$ необхідно звести до нормального вигляду.

Розглянемо для додаткового обмеження схему структурного графу конфігурації перестановок, враховуючи що для дробово-лінійної функції вибраний метод генерування конфігурації перестановок – метод переміщення максимального елемента. Тоді схема структурного графу будується, враховуючи специфіку алгоритму методу переміщення максимального елементу: в крайніх екстремальних вершинах схеми розміщені точки для яких місце максимального елементу зафіксоване в залежності від рівня підграфа, а останні координати цих точок упорядковані, враховуючи те,що якщо має місце $c_1 \leq c_2 \leq \ldots \leq c_n$, то максимум функції $F(x,c,u)$ досягається для перестановки $u_0 = (1,2,\ldots,n)$, а мінімум – для перестановки $u^* = (n,n-1,\ldots,2,1)$.

Тоді структурна схема графу перестановок має вигляд:

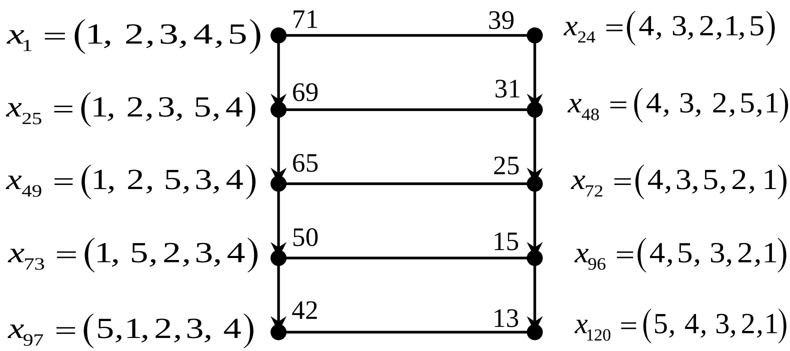


Рис. 6.20. Структурна схема графу перестановок для g_1

Згідно рис. 6.20 умову додаткового обмеження задовольняють частково точки з схеми підграфів G_1, G_2. Тому для них необхідно також зробити розклад за схемами підграфів.

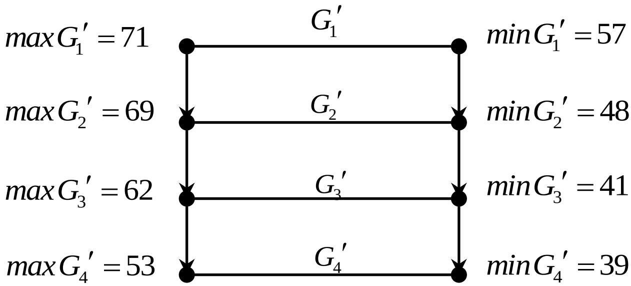


Рис. 6.21. Структурна схема підграфів для підграфа G_1

Тоді на основі подальших розрахунків на підграфах G_1', G_2' визначимо необхідні точки:

№	x_1	x_2	x_3	x_4	x_5	g_1
1	1	2	3	4	5	71
2	1	2	4	3	5	69
3	1	3	2	4	5	66
7	2	1	3	4	5	69
8	2	1	4	3	5	67

Здійснимо обернене перетворення $u^{-1} = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 1\,2\,3\,4\,5 \end{pmatrix}$ і отримаємо наступні точки, що визначають область визначення функції:

№	x_1	x_2	x_3	x_4	x_5	g_1
1	4	5	1	2	3	71
2	3	5	1	2	4	69
3	4	5	1	3	2	66
7	4	5	2	1	3	69
8	3	5	2	1	4	67

Наступним кроком є визначання максимального значення функції в цих точках. Скористаємося відомим розкладом за підграфами значеннями дробово-лінійної функції і порівнявши результати, отримаємо вершину (3 5 2 1 4), що доставляє максимальне значення цільової функції $F(x) = \dfrac{f(x)}{g(x)}$.

Висновки до розділу

Досліджено складні задачі з лінійною та дробово-лінійною цільовими функціями і додатковими лінійними обмеженнями на комбінаторних конфігураціях і застосовано загальну схему методу направленого структурування до розв’язування таких задач.

Зокрема, метод направленого структурування, застосований до екстремальних задач з лінійною функцією цілі та додатковими лінійними обмеженнями на комбінаторних конфігураціях перестановок, розміщень, сполучень. Також даний метод дістав подальший розвиток для задач з дробово-лінійними функціями цілі та додатковими лінійними обмеженнями на комбінаторній множині перестановок.

В розділі представлено ряд числових експериментів, які підтверджують ефективність методу направленого структурування для розв’язування екстремальних задач на різних комбінаторних конфігураціях та для різних цільових функцій.

Моделі екстремальних задач з додатковими лінійними обмеженнями та комбінаторними властивостями області допустимих розв’язків можуть бути застосована для ухвалення рішень в практичних задачах економіки, техніки, народного господарства.

РОЗДІЛ 7. ДОСЛІДЖЕННЯ ТА РОЗВ'ЯЗАННЯ ЕКСТРЕМАЛЬНИХ ЗАДАЧ ОПТИМІЗАЦІЇ ПРИ УМОВІ БАГАТОКРИТЕРІАЛЬНОСТІ НА КОМБІНАТОРНИХ КОНФІГУРАЦІЯХ

Екстремальні задачі оптимізації декількох функцій виникають при дослідженні багатьох теоретичних і прикладних проблем. Практично будь-яка прикладна задача вимагає, щоб шуканий розв'язок знаходився з урахуванням багатьох критеріїв. Останні, як правило, суперечливі в тому розумінні, що якість порівнюваних альтернатив неможливо адекватно виразити одним комплексним критерієм, що представляє деяку згортку вхідних критеріїв.

Задача ускладнюється, якщо накладаються додаткові умови комбінаторності, тобто задача розглядається на деякій комбінаторній конфігурації. Це означає, що апарат класичної оптимізації є недостатнім для пошуку і прийняття ефективних розв'язків.

Існування оптимізаційних задач з багатьма критеріями на комбінаторних конфігураціях вимагає розробки ефективних методів та алгоритмів оптимізації, які дозволяють розв'язувати як окремі задачі, так і цілі їх класи. Це робить необхідним вивчення властивостей указаних оптимізаційних задач, що в свою чергу приводить до дослідження властивостей множин їх розв'язків та областей визначення. Таким чином, виникає необхідність в використанні, подальшому дослідженні та розвитку більш широкої і більш загальної теорії та розробці нових методів розв'язання такого класу задач.

Дослідження в області багатокритеріальної оптимізації в даний час особливо інтенсивно стимулюються практичними потребами і розвитком комп'ютерних інформаційних технологій. Тому з'явилась велика кількість праць, присвячена задачам багатокритеріальної оптимізації [31, 33, 42, 64, 65, 66–85, 88, 101–105, 108, 113–119, 136–139, 143–145, 159, 166–171, 173, 185–189, 209–212, 221–224, 280, 295, 308, 319, 323, 330]. Задачі з багатьма критеріями можна розглядати як умовні так і безумовні. В попередніх розділах зроблено задальну постановку екстремальної безумовної задачі на комбінаторних конфігураціях та запропоновано новий метод направленого структурування, що ґрунтується на використанні графів комбінаторних конфігурацій.

Даний розділ присвячено дослідженню математичних моделей і методів розв’язання екстремальних задач з урахуванням багатокритеріальності. Особливістю цих задач є оцінка якості розв’язку за наявності декількох, як правило несумісних критеріїв. Залежно від постановки задачі, різні критерії входять до складу векторної цільової функції в різних комбінаціях, породжуючи тим самим різні типи задач багатокритеріальної оптимізації.

В умовах багатокритеріальності вибір найдоцільнішого розв’язку здійснюється з множини векторно-непорівнюваних, конкуруючих альтернатив. Створення теоретично обгрунтованих методів дослідження множини альтернатив з урахуванням властивостей цих множин дозволяє одержувати раціональні підходи до їх розв’язання. Реалізація цих методів на комп’ютері значно зменшує витрати в процесі ухвалення рішення. Розв’язки, одержані цими методами, з урахуванням взаємозв’язку критеріїв, приводять до істотної економії засобів що витрачаються на них і ресурсів. Тому проблема відшукання множини альтернатив для багатокритеріальних задач та побудова нових методів для їх знаходження має велике практичне і теоретичне значення та потребує подальшого дослідження і удосконалення.

Розглянемо постановку екстремальної задачі комбінаторної оптимізації, що об’єднує проблему багатокритеріальності та комбінаторні властивості розв’язків з додатковими умовами.

7.1. Математична постановка екстремальної задачі за умови багатокритеріальності та метод її розв’язання

Визначимо екстремальну багатокритеріальну задачу як обчислювальну проблема, у якій задана множина альтернатив $X=\{x\}$, векторна цільова функція $F(x): X \to R,$ де $F(x)=\left(f_1(x),\ldots, f_i(x),\ldots, f_l(x)\right),$ що складається із часткових критеріїв $f_i(x) \to max,\ i \in N_l$. Потрібно знайти альтернативу

$x^0 \in X$, на якій ця цільова функція приймає екстремальне значення: $F\left(x^0\right) = \underset{x \in X}{extr}\, F\left(X\right)$, $extr \in \{min, max\}$. Альтернативи $x \in X$ називають допустимими розв'язками, x^0 – оптимальний розв'язок, $X = \{x\}$ – множиною допустимих розв'язків. Тоді $X = \{x\}$ – визначається деякою комбінаторною конфігурацією, елементи якої генеруються новими методами, що запропоновані в другому розділі.

Зокрема, в залежності від властивостей задачі – вигляду часткових критеріїв, додаткових обмежень, використовуються методи генерування комбінаторних об'єктів: рекурсивний, метод переміщення максимального елементу.

Як відомо, одним з основних, фундаментальних понять багатокритеріальної оптимізації взагалі є поняття оптимального по Парето, тобто ефективного розв'язку та оптимального по Слейтеру – слабо ефективного розв'язку.

Визначення 7.1. Допустимий розв'язок $x^0 \in X$ розглядуваної багатокритеріальної задачі називається ефективним або Парето-оптимальним, якщо не існує такого розв'язку $x \in X$, що $F(x) \geq F\left(x^0\right)$, $F(x) \neq F\left(x^0\right)$, тобто для якого мають місце нерівності

$$f_k(x) \geq f_k\left(x^0\right),\ k \in N_l,$$

причому хоча б одна з цих нерівностей строга.

Головним завданням є відшукання ефективного розв'язку задачі.

Серед методів пошуку Парето-оптимальних розв'язків найбільшого розповсюдження отримали так звані алгоритми лінійної згортки. Ці алгоритми базуються на наступному відомому факті: при визначеній цільовій вектор-функції елемент, що мінімізує (максимізує) лінійну згортку $\sum_{i=1}^{l} \mu_i F\left(x_i\right)$, є Парето-оптимальними. З огляду на специфіку області допустимих розв'язків екстремальної багатокритеріальної задачі на комбінаторній конфігурації, слід зазначити, що функція:

$$F(x)=\left(f_1(x),\ldots,f_l(x)\right)=\sum_{i=1}^{l}\mu_i f\left(x_i\right)$$

є зростаючою за відношенням $\geq$ на деякій комбінаторній конфігурації, якщо всі числа $\mu_1,\ \mu_2,\ \ldots,\ \mu_l$ додатні. У випадку, якщо який-небудь з коефіцієнтів $\mu_i=1$, а всі інші, $\mu_j=0,\ i\neq j,$ $i,j\in N_l$, то розглядається однокритеріальна задача з i-ю цільовою функцією. Отже, максимізація функції $\sum_{i=1}^{l}\mu_i f\left(x_i\right)$ на комбінаторній конфігурації приводить до знаходження Парето-оптимальних розв'язків.

Продовжуючи дослідження і розвиваючи отримані в попередніх розділах результати, запропонований підхід до розв'язання задачі $Z\left(F(x),X\right)$, де $F(x)$ – вектор-функція, X – множина допустимих розв'язків, оснований на лінійній згортці (агрегації) часткових критеріїв задачі і подальшому зведенні пошуку розв'язку початкової задачі до розв'язку серії скалярних (однокритеріальних) задач, перевірки оптимальності отриманих розв'язків. Методи розв'язків однокритеріальних задач основані на ідеях декомпозиції, відтинаючих площин Келлі, релаксації та методі направленого структурування.

Для розв'язування екстремальних багатокритеріальних задач застосуємо новий метод направленого структурування, який описаний і реалізований для розв'язування однокритеріальних екстремальних задач в попередніх розділах.

Підхід до розв'язування багатокритеріальної екстремальної задачі на основі методу направленого структурування полягає у наступному:

1) багатокритеріальна задача зводиться до однокритеріальної за допомогою лінійної згортки: для цього задаються вагові додатні коефіцієнти $\mu_j, j\in N_l$, які визначають ступінь важливості кожного критерію, і максимізується лінійна комбінація цільових функцій, тобто розв'язується задача:

$$Z\left(f, G^s\right),$$

де $f(x)=\sum_{i=1}^{l}\mu_i\langle c_i, x\rangle,\ \mu_i\geq 0,\ i\in N_l,\ \sum_{i=1}^{l}\mu_i=1,\ G^s=R^n$.

2) генеруються у певній послідовності всі елементи заданої комбінаторної конфігурації;

3) будується на її елементах орієнтований граф, де дуга відповідає спаданню значень цільової функції;

4) будується поліноміальний алгоритм для розв’язування задачі на частково упорядкованих вершинах графа;

5) визначається множина Парето-оптимальних розв’язків.

Дослідимо властивості області допустимих розв’язків.

7.2. Властивості області допустимих розв’язків багатокритеріальних задач

В попередньому пункті розглянуто означення Парето-оптимального розв’язку. Крім цих розв’язків розглянемо розв’язок оптимальний по Слейтеру – слабо ефективний.

Визначення 7.2. Розв’язок $x^0\in X$ слабо ефективний, ефективний по Слейтеру, якщо не існує такого розв’язку $x\in X$, що $x>x^0$, тобто для якого мають місце нерівності

$$F_k(x)>F_k\left(x^0\right), k\in N_n.$$

Множину ефективний розв’язків будемо позначати через $P_F(X)$, а слабо ефективних через $Sl_F(X)$.

Розрізняють також множину строго ефективних розв’язків: розв’язок $x^0\in X$ строго ефективний, ефективний по Смейлу, якщо не існує такого розв’язку $x\in X$, що $x>x^0$, $x\neq x^0$ тобто для якого мають місце нерівності

$$F_k(x)\geq F_k\left(x^0\right), k\in N_n.$$

Відповідно множину строго ефективних розв'язків позначають $Sm_F(X)$ і для всіх вище зазначених множин, згідно [172, 192, 206, 207, 210, 223] виконується співвідношення

$$Sm_F(X) \subseteq P_F(X) \subseteq Sl_F(X). \tag{7.1}$$

В багатокритеріальній задачі кожний розв'язок $x \in X$ повністю характеризується своєю оцінкою $y = F(x)$, і тому вибір оптимального розв'язку зводиться до вибору оптимальної оцінки із множини Y всіх допустимих оцінок. Тоді є важливим задача порівняння за перевагою векторних оцінок, тобто значення векторного критерію $F(x) = (F_1(x), F_2(x), \dots, F_n(x))$. Зрозуміло, що найбільш просто співставити за перевагою ті векторні оцінки, які відрізняють одна від одної лише однією компонентою. Тому інформація за перевагою змін значень одного часткового критерію при фіксованих значеннях всіх останніх критеріїв є найбільш доступною і достовірною, а тому як раз таку інформацію доцільно одержувати в першу чергу і використовувати для аналізу задачі.

Розроблено багато різних принципів прийняття рішень в ситуаціях такого роду. Але найбільш цікавими є традиційні, які пов'язані із виділенням із всієї множини розв'язків та оцінок $Y = \{y = \Phi(a) \mid a \in E\}$ множини непокращуваних або оптимальних по Парето, оптимальних по Слейтеру, оптимальних по Смейлу векторів.

Для подальшого викладу матеріалу розглянемо екстремальну багатокритеріальну задачу на комбінаторній конфігурації перестановок, враховуючи, що:

$$M_n(A) = conv\, P_n(A),$$

згідно [87, 239, 242], будемо розглядати елементи множини перестановок як вершини графа многогранника перестановок, вигляд якого відоми і представлений в другому і третьому розділах для однокритеріальних екстремальних задач як орієнтований граф за значеннями лінійної цільової функції.

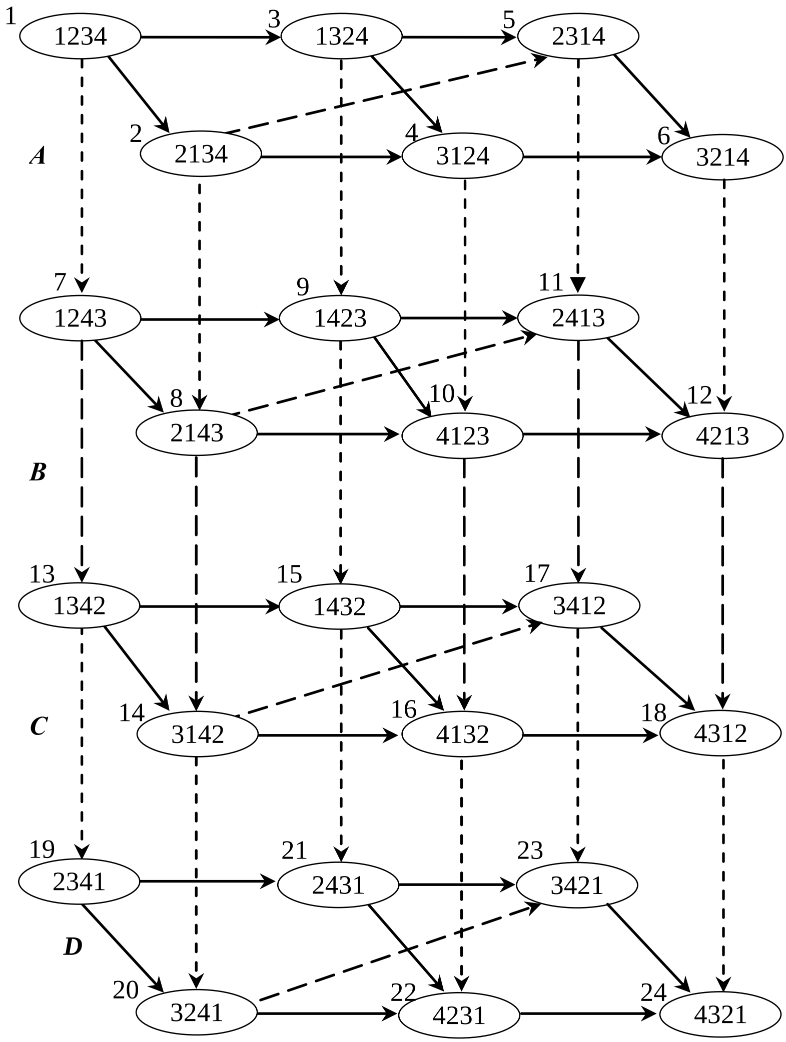


Рис. 7.1. Представлення графа перестановок $G(P_n)$ для $n=4$

Тоді можна сформулювати задачу $Z(F,X)$ максимізації деякого векторного критерію $F(x)$ на множині *X,* причому кожній точці $a \in P_n(A)$ буде відповідати точка $x \in X$, така, що $F(x) = \Phi(a)$.

$$Z(F,X): max\{F(x) \mid x \in X\},$$

де $F(x) = (f_1(x), f_2(x), \ldots, f_l(x))$, $f_i : R^n \to R^1, i \in N_l$, X – непорожня множина у R^n, що визначається таким чином $X = vert\, M_n(A)$.

Оскільки для багатокритеріальних екстремальних задач на першому етапі метод направленого структурування зводить задачу до однокритеріальної, то зрозуміло, що можна використовувати орієнтований граф, представлений на рис. 7.1 для подальшого знаходження розв’язку багатокритеріальної задачі.

Для комбінаторних багатокритеріальних задач розв’язок представляє узагальнення поняття точки максимуму числової функції: розв’язок Парето-оптимальний, якщо значення кожного із критеріїв можна поліпшити лише за рахунок погіршення значень інших критеріїв.

Властивостям і методам відшукання Парето-оптимальних розв’язків присвячено досить багато літератури, але для задачі $Z(F,X)$ комбінаторної оптимізації необхідно врахувати специфіку й комбінаторні властивості області допустимих розв’язків, тому є актуальним розглянути дане питання.

З множини розв’язків *X* необхідно вибрати такі, для яких виконувалася б умова належності комбінаторній конфігурації перестановок $x \in P_{nk}(A)$ і які були б «кращі» ніж інші. Визначимо оптимальні розв’язки, тобто такі, які мають переваги над множиною інших розв’язків. Прямий перебір деякого найкращого розв’язку, як правило є трудомістким. Тому визначемо правило вибору найкращого розв’язку з двох даних.

Визначення 7.3. Якщо з двох заданих розв’язків x_1 і x_2 множини *X*, вибирається розв’язок x_1, то розв’язок x_1 переважає над розв’язком x_2.

Всі пари вигляду (x_1, x_2), де x_1, $x_2 \in X$, для яких розв’язок x_1 переважає, над розв’язком x_2, утворюють деяку множину.

Відношення строгої переваги, що використовується при побудові множини позначимо символом $\succ$.

Відношення $\succ$ повинне бути іррефлексивним, також слід вважати відношення $\succ$ асиметричним, оскільки інакше можуть одночасно виконуватися співвідношення $x_1 \succ x_2$ і $x_2 \succ x_1$, що суперечить означенню.

У багатьох випадках доцільно припускати введене відношення $\succ$ ще і транзитивним. Транзитивність відношення означає, що розв'язок x_1 переважає перед x_2, а $x_2 \succ x_3$, тобто x_2 переважає над x_3, то з двох розв'язків x_1 і x_2 буде вибрано x_1.

Множину всіх оптимальних розв'язків множини X позначатимемо через $opt_{\succ} X$. Залежно від структури X і виду відношення $\succ$ множина $opt_{\succ} X$ може містити єдиний елемент, скінченну або нескінченну множину елементів, а також не містити жодного елементу. Якщо врахувати комбінаторну природу множини допустимих розв'язків задачі $Z(F,X)$, то $opt_{\succ} X$ – є скінченною множиною, елементи якої є точки вибраної комбінаторної конфігурації, зокрема розміщень, перестановок, тобто $x \in P_{nk}$, чи $x \in A_{qk}^{n}$ та ін.

Отже для задачі $Z(F,X)$ множина $opt_{\succ} X$ містить принаймні два елементи. Розглянемо два довільні оптимальні розв'язки x_1 і x_2. Оскільки передбачено, що розв'язки є оптимальними, то для них не може виконуватися відношення переваги $\succ$. Таким чином, згідно зазначеної умови, для розв'язків може виконуватися лише одне з трьох наступних співвідношень: $x_1 \succ x_2$, або $x_2 \succ x_1$, або рівні один одному.

Сформулюємо теорему, яка гарантує існування оптимальних розв'язків для комбінаторних багатокритеріальних задач оптимізації $Z(F,X)$.

Теорема 7.1. Так як множина допустимих розв'язків $X = vert\, M(A) \cap D$ задачі $Z(F,X)$ не порожня і містить скінченне число елементів, а відношення $\succ$ асиметричне і транзитивне, то множина оптимальних розв'язків непуста, тобто $opt \succ X \neq \varnothing$.

Доведення цього твердження носить конструктивний характер і його можна сформулювати у вигляді алгоритму [191]:

Введемо позначення $X = X_1 = \left\{x_{11},\ x_{12},\ \ldots,\ x_{1n_1}\right\}$.

Нехай $n_1 = 1,$ то $X_1 = \left\{x_{11}\right\} = opt_{\succ} X_1$. Тому далі вважатимемо $n_1 > 1$.

Перший крок алгоритму полягає в попарному порівнянні розв'язку x_{11} з кожним з решти розв'язків множини. Якщо для деякого $i \in \left\{2, 3,\ \ldots,\ n_1\right\}$ виконується співвідношення $x_{11} > x_{1i}$, то розв'язок x_{1i} з множини X_1 видаляють; і він не може бути оптимальним. Інакше, тобто коли $x_1 \geq x_{1i}$, або $x_{1i} > x_{11}$ то розв'язок x_{1i} зберігають. Якщо ні для якого $i = 2, 3, \ldots, n_1$ не виконалось співвідношення $x_{1i} > x_{11}$, то розв'язок x_{11} є оптимальним і його потрібно запам'ятати. Множину розв'язків, що залишилася в результаті вилучення, позначимо через:

$$X_2 = \left\{x_{21}, x_{22}, \ldots, x_{2n_2}\right\}, n_2 < n_1.$$

Якщо $X_2 = \varnothing,$ то розв'язок x_{11} оптимальний (він зберігається в пам'яті), оскільки через асиметричність відношення $\succ$ із співвідношення $x_{11} \succ x_{1i}$ випливає, що $x_{1i} > x_{11}, i = 2, 3, \ldots, n_1,$ не може мати місця. В такому випадку процедура відшукання множини $opt_{\succ} X_1$ закінчена. Якщо ж $X_2 \neq \varnothing,$ то переходимо до наступного кроку алгоритму.

Другий крок аналогічний першому і полягає в попарному порівнянні розв'язку x_{21} з кожним із розв'язків $x_{22}, \ldots, x_{2n_2}$. Всі розв'язки x_{2i}, для яких виконується відношення переваги $x_{21} > x_{2i}$ для x_{21} із множини X_2 виключають. Крім того, видаляють розв'язок x_{21}. При цьому, якщо ні для якого $i = 2, 3, \ldots, n_2$ не виконується співвідношення $x_{2i} > x_{21}$, то $x_{21} \in opt_{\succ} X_2$. Більш того $x_{21} \in opt_{\succ} X_1$ то розв'язок x_{21} слід запам'ятати. Насправді, співвідношення $x_{11} \succ x_{21}$ не може мати місця, оскільки розв'язок x_{21} не був видалений з X_1 на першому кроці. Співвідношення $x_{1i} \succ x_{2i}$ для $x_{1i} \in X_1 \setminus X_2, i \neq 1$ також не може бути виконано, оскільки $x_{11} \succ x_{1i}$ і відношення $\succ$ транзитивне: з $x_{11} \succ x_{1i}$ і $x_{1i} \succ x_{21}$ слідує, що $x_{11} \succ x_{21}$. Множину розв'язків, що залишилася, після вилучення позначимо $X_3 = \left\{x_{31}, x_{32}, \dots, x_{3n_3}\right\}, n_3 < n_2$. Якщо $X_3 \neq 0$, то переходимо на наступний крок і т. д.

Алгоритм, згідно транзитивності відношення $\succ$, дає можливість знайти розв'язок x_{k1}, оптимальний на множині X_k, тобто $opt_{\succ} X_1$, а значить, і на початковій множині $X = vertM\left(A\right) \cap D$.

Оскільки множина $X = vertM\left(A\right) \cap D$ містить скінченне число елементів, то через скінчене число кроків процедура закінчить свою роботу. Розв'язки, що зберігаються в пам'яті, утворюють шукану непусту множину $opt_{\succ} X$.

Можна оцінити «трудомісткість» сформульованого алгоритму, тобто визначити якнайменше і найбільш можливе число попарних порівнянь, які потрібно зробити для знаходження всієї множини $opt_{\succ} X$.

Найменше число порівнянь $n_1 - 1$ має місце, якщо $x_{11} \succ x_{1i}$, $i = 2, 3, \dots, n_1$. У «найдовшому варіанті» доведеться порівнювати між собою всі можливі пари розв'язків, і тому максимальне число порівнянь рівне $n_1\left(n_1 - 1\right)/2$.

Теорема 7.2. Елементи множин $P_F\left(X\right)$ – Парето-оптимальних, $Sl_F\left(X\right)$ – слабо ефективних розв'язків багатокритеріальної комбінаторної задачі вигляду $Z\left(F, X\right)$ на перестановках P_n знаходяться у вершинах графа $G\left(A\right)$ переставного многогранника $M_n\left(A\right)$.

Доведення. Враховуючи співвідношення (7.1) між введеними множинами ефективних розв'язків і той факт, що множина допустимих розв'язків X є підмножиною множини перестановок P_n, справедливе наступне співвідношення:

$$P_F\left(X\right) \subseteq Sl_F\left(X\right) \subset P_n\left(A\right). \tag{7.2}$$

Як уже було зазначено, відповідно до [242, 245] множина перестановок P_n збігається з множиною вершин загального переставного многогранника $vert M_n(A)$. Таким чином, справедливе включення (7.2). Теорема доведена.

Нехай функції $f_i(x), i \in N_l$, векторного критерію $F(x)$ є лінійними, тобто $f_i(x) = \langle c_i, x \rangle, i \in N_l$. Важливі властивості допустимої області X і множин різних видів ефективних розв'язків, зазначені в теоремі 7.2, а також лінійність функцій векторного критерію дозволяють звести розв'язання задачі $Z(F,X)$ до розв'язання задачі $Z(F,G)$ визначеній на допустимій множині $G = M \bigcap D$.

При встановленні різних видів ефективності розв'язків, якщо виконуються необхідні умови оптимальності розглянутого розв'язку, то гарантувати його ефективність не можна, однак, якщо ці умови не виконуються, то даний розв'язок не ефективний. Якщо використовуються достатні умови, то розв'язок, що їх задовольняє, ефективний, у протилежному випадку питання про ефективність розв'язків залишається відкритим. Якщо ж застосовуються необхідні і достатні умови, то питання вирішується однозначно: розв'язок ефективний тоді і тільки тоді, коли він задовольняє цим умовам.

Як було зазначено, наряду з множиною різних видів оптимальності розв'язків є доцільним розглянути поняття оптимальних оцінок багатокритеріальних комбінаторних задач, які відіграють важливу роль у теорії багатокритеріальної оптимізації.

Одним із основних понять Парето-оптимального розв'язку багатокритеріальних задач є поняття оцінки. Тобто, у багатокритеріальній задачі кожний розв'язок $x \in X$ повністю характеризуються своєю оцінкою $y = (y_1, \ldots, y_l)$, де $y_i = f_i(x)$, $i \in N_l$. Тому вибір оптимального розв'язку зводиться до вибору оптимальної оцінки з множиною Y всіх досяжних оцінок.

Множина оцінок для екстремальної багатокритеріальної задачі оптимізації $Z(F,X)$ визначається таким чином:

$$Y = F(X) = \left\{ y \in R \middle| y = F(x), x \in E \right\}.$$

Отже, вибір розв'язку з елементів комбінаторної конфігурації E рівносильний вибору відповідної оцінки з Y. Розглянемо властивості Парето-оптимальних оцінок.

Так як множина Парето-оптимальних розв'язків у багатокритеріальних комбінаторних задача позначена $P_F(X)$, тоді множину Парето-оптимальних оцінок позначимо $P(F)$.

У багатокритеріальних задачах порівнюються по перевазі векторні оцінки, тобто значення векторного критерію $F(f(x)) = (f_1(x), f_2(x), \ldots, f_l(x))$.

Наряду з множиною Парето-оптимальних розв'язків екстремальної багатокритеріальної задачі розглядається множина парето-оптимальних векторів в наступних роботах [172, 192].

Визначення 7.4. Вектор $F(x^*)$ для Парето-оптимального розв'язку x^* називають Парето-оптимальним вектором розв'язку або Парето-оптимальною оцінкою, а множина всіх таких векторів – множиною Парето-оптимальних векторів або оцінок.

Позначимо

$$P(F) = f(P_F(X)) = \left\{ F(x^0) \in Y \mid \exists x^0 \in P_F(X) \right\},$$

де Y означає множина можливих векторів, тобто $Y = F(x)$.

Природно, що найбільш просто порівнювати по перевазі ті векторні оцінки, які відрізняються один від одного лише однією компонентою [117, 170, 189]. Тому інформація про переваги зміни значення одного приватного критерію при фіксованих значеннях всіх інших критеріїв є найбільш доступна і достовірна, і саме її доцільно визначати в першу чергу й використати для аналізу задачі, але такі ситуації бувають рідко, тому є необхідним більш глибоко досліджувати структуру області допустимих розв'язків.

Для подальшого розгляду поняття Парето-оптимального розв'язку й оцінки комбінаторних багатокритеріальних задач оптимізації сформулюємо:

Твердження 7.1. Максимум лінійної функції однокритеріальної комбінаторної задачі:

$$f(x) = \sum_{j=1}^{k} c_j x_j, \tag{7.3}$$

де $c_{\alpha_1} \geq ...c_{\alpha_s} \geq 0 > c_{\alpha_{s+1}} \geq ... \geq c_{\alpha_k}$, $s \in N_k$, $\alpha \in N_k$, на комбінаторній множині E (перестановок, розміщень) досягається в точці $x^* = \left(x_1^*,...,x_k^*\right) \in E$, що задовольняє умовам:

$$x^*_{\alpha_i} = a_i \forall i \in N_s; \; x^*_{\alpha_{s+i}} = a_{\eta - r + i} \forall i \in N_r,$$

якщо елементи мультимножини A впорядковані в такий спосіб:

$$a_{i_1} \geq a_{i_2} \geq \ldots \geq a_{i_k}, \tag{7.4}$$

де r, s-константи, що задовольняють умовам $r + s = k$, $r, s \in N_k$, а мінімум лінійної функції однокритеріальної комбінаторної задачі $f(x) = \sum_{j=1}^{k} c_j x_j$, в точці $x^* = \left(x_1^*, \ldots, x_k^*\right) \in E$, що задовольняє умовам:

$$x^*_{\alpha_i} = a_i \forall i \in N_s; \; x^*_{\alpha_{s+i}} = a_{\eta - r + i} \forall i \in N_r,$$

якщо елементи мультимножини A впорядковані таким способом:

$$a_{i_1} \leq a_{i_2} \leq \ldots \leq a_{i_k},$$

де r, s-константи, що задовольняють умовам $r + s = k$, $r, s \in N_k$.

Позначимо x^0 – оптимальний розв'язок, то $f\left(x^0\right) = y^0$ – оптимальна оцінка даного розв'язку, тоді $F\big(f(x)\big) = \big(f_1(x), f_2(x), \ldots, f_l(x)\big)$ можна записати таким способом $F(y) = (y_1, y_2, ..., y_l)$.

Визначення 7.5. Функція $F(y) = (y_1, y_2, \ldots, y_l)$, що визначена на множині оцінок $Y \subset R^m$, є зростаючою по відношенню $\geq$, якщо з виконання нерівності $y \geq y'$ для векторів $y, y' \in Y$, завжди виконується нерівність $F(y) \geq F(y')$.

Це означення являє собою узагальнене поняття зростаючої функції однієї змінної на випадок функції багатьох змінних.

Теорема 7.3. Якщо функція $F(y)$ зростає по відношенню $\geq$ і x^0 – точка максимуму функції $F\left(f(x)\right)= =F\left(f_1(x), f_2(x),\ldots, f_l(x)\right)$ на комбінаторній конфігурації перестановок, то $F\left(x^0\right)\in P(A)$, що означає $x^0 \in P_F(X)$.

Доведення. Якщо x^0 – точка максимуму функції $F\left(f(x)\right)=F\left(f_1(x), f_2(x),\ldots, f_l(x)\right)$, то відповідно до означення Парето-оптимального розв'язку, вона належить множині Парето-оптимальних розв'язків, а по означенню Парето-оптимальних оцінок виконується наступне співвідношення $f\left(x^0\right)\in P_y(F,X)$.

Теорема 7.3 сформульована в термінах розв'язків. Але її можна переформулювати у термінах оцінок.

Теорема 7.4. Нехай функція $F(y)$ визначена на множині оцінок $Y\subset R^m$. Для того, щоб точка $y^0\in Y$ була Парето-оптимальною оцінкою, тобто $y^0\in P(F)$ для екстремальної багатокритеріальної задачі оптимізації $Z(F,X)$, достатньо, щоб вона була точкою максимуму на множині Y функції $F(y)$, що зростає по відношенню $\geq$, де Y – множина оцінок.

Доведення. Доведення випливає з теореми 7.3. Нехай, за умовою $y^0\in Y$ і $F\left(y^0\right)\geq F(y)$ для всіх $y\in Y$. Припустимо обернене: що для деякої оцінки $y'\in Y$ виконується нерівність $y'\geq y^0$. Звідси, оскільки функція F зростаюча, одержуємо нерівність $F(y')>F\left(y^0\right)$, що суперечить попередній. У свою чергу, якщо максимум досягався в точці, що належить вершині многогранника, тобто $x^0\in M$, то вона є Парето-оптимальним розв'язком задачі $Z(F,X)$, згідно теореми 7.3 і співвідношення $F\left(x^0\right)=y^0$, тоді точка $y^0\in P(F)$. Теорема доведена.

Серед методів відшукання Парето-оптимальних розв'язків найбільшого розповсюдження одержали так звані алгоритми лінійної згортки. Ці алгоритми базуються на наступному відомому факті: при додатньо-визначеній цільовій вектор-функції елемент, що мінімізує (максимізує) лінійну згортку $\sum_{i=1}^{l}\mu_i F(x_i)$, є Парето-оптимальними. З огляду на специфіку області допустимих розв'язків комбінаторної задачі, слід зазначити, що функція $F(x)=\left(f_1(x),\ldots,f_l(x)\right)=\sum_{i=1}^{l}\mu_i f(x_i)$ є зростаючою по відношенню $\geq$ на комбінаторній конфігурації розміщень, перестановок, якщо всі числа $\mu_1,\mu_2,\ldots,\mu_n$ додатні. Отже, максимізація функції $\sum_{i=1}^{l}\mu_i f_i(x)$ на комбінаторній множині E (розміщень, перестановок) приводить до знаходження Парето-оптимального розв'язків.

Зокрема, для багатокритеріальних безумовних задач на комбінаторній конфігурації перестановок встановлено наступний факт в вигляді теореми.

Теорема 7.5. Розв'язок $x^0\in S$ комбінаторної багатокритеріальної задачі $Z(F,S)$ є Парето-оптимальним, якщо існують числа $\mu_i\geq 0, i\in N_l$, $\sum_{i=1}^{l}\mu_i=1$ такі, що максимізують лінійну згортку

$$\sum_{i=1}^{l}\mu_i f_i\left(x^0\right)=\max_{x\in S}\sum_{i=1}^{l}\mu_i f_i(x)$$

критеріїв $f_1(x), f_2(x),\ldots, f_l(x)$ на заданій комбінаторній конфігурації перестановок.

Доведення. Множина допустимих розв'язання X задачі багатокритеріальної комбінаторної оптимізації $Z(F,X)$ є скінченою і обмеженою, тому що $X=vertM(A)\cap D$. Тоді, відповідно до твердження, існує розв'язок $x^0\in X$ задачі $Z(F,X)$, який є оптимальним, а відповідно Парето-оптимальним для комбінаторної багатокритеріальної задачі. Теорема доведена.

Теорема 7.5 показує, що при деяких припущеннях, підбираючи коефіцієнти $\mu_1,\mu_2,\ldots,\mu_l$, будь-який Парето-оптимальний розв'язок можна одержати в результаті розв'язання відповідної однокритеріальної задачі максимізації. Аналогічний висновок справедливий і для оцінок.

Твердження 7.1. Якщо компоненти вектор-функції

$$F(x)=\left(f_1(x),\ldots,f_l(x)\right)$$

неперервні на деякій комбінаторній конфігурації, то $P(F)\neq\varnothing$.

Справедливість даного твердження очевидно.

Твердження 7.2. Для комбінаторної багатокритеріальної задачі $Z(F,X)$ існує хоча б один Парето-оптимальний розв'язок й, відповідно, хоча б один Парето-оптимальний вектор, тобто $P_F(X)\neq\varnothing,\ P(F)\neq\varnothing$.

З вище сформульованих теорем і тверджень випливає наступний спосіб відшукання одного з множини Парето-оптимальних розв'язків: знайти точку, що реалізує максимальне значення функції $\sum_{i=1}^{l} f_i(x)$ на множині допустимих розв'язання X [191, 192].

Враховуючи вище сказане, можна визначити умови знаходження розв'язку багатокритеріальної задачі $Z(F,X)$ без додаткових обмежень. Таку багатокритеріальну задачу назвемо багатокритеріальною безумовною задачею з лінійними критеріями на комбінаторній конфігурації. Тоді можна сформулювати наступне твердження.

Твердження 7.3. Якщо P_i – множина перестановок $\alpha^i=\left(\alpha_1^i,\ldots,\alpha_k^i\right)$ елементів множини перших натуральних чисел, що задовольняють умові:

$$c^i_{\alpha^i_1}\geq c^i_{\alpha^i_2}\geq\ldots\geq c^i_{\alpha^i_s}\geq 0> c^i_{\alpha^i_{s+1}}\geq\ldots\geq c^i_{\alpha^i_k},$$

то множина абсолютних розв'язків задачі $Z(F,X)$ не порожня, якщо не порожня множина, що є перетином множини розв'язків, знайдених для всіх критеріїв даної задачі:

$$\bigcap_{i=1}^{l} R_i = R\neq\varnothing.$$

Розглянуте вище твердження не дає відповіді на питання: які розв'язки знайдені – ефективні, слабо ефективні або строго ефективні.

Сформулюємо наступне твердження.

Твердження 7.4. Оптимальні розв'язки багатокритеріальних комбінаторних задач на комбінаторних конфігураціях знаходяться у вершинах графа многогранника, де для множини перестановок $P_n = vert\, M_n\left(A\right)$, і визначають Парето-оптимальну множину.

Доведення. Якщо розглядати багатокритеріальну комбінаторну задачу $Z\left(F, X\right)$ без додаткових обмежень, то згідно твердження 7.5 оптимальний розв'язок – це точки комбінаторної конфігурації, що визначаються як вершини графа. Як відомо $vert\, M_{nk}\left(A\right) = P_{nk}$, то при накладенні додаткових умов *D*, ефективні розв'язки також будуть у вершинах, оскільки умови, які утворюють многогранну множину *D* тільки звужують область, яка описує область допустимих розв'язків $X = vert\, M \cap D$. Зрозуміло, що випадок $X = \varnothing$ не розглядається.

Теорема доведена.

Завдяки наявності вказаного вище прямого зв'язку між множиною ефективних розв'язків і Парето-оптимальних векторів та описаних вище властивостей можна застосувати алгоритм знаходження множини Парето-оптимальних векторів для задачі комбінаторної оптимізації $Z\left(F, X\right)$, аналогічний [119].

Алгоритм

Крок 1. Для всіх можливих точок $x = \left(x_1, \ldots, x_n\right) \in P_n$, де P_n – комбінаторна конфігурація перестановок визначаємо множину можливих векторів $Y = \left\{y^1, y^2, \ldots, y^l\right\}$.

Крок 2. Утворюємо множину Парето-оптимальних векторів, вибравши $P\left(Y\right) = f\left(P_f\left(X\right)\right)$ ті, які співпадають з множиною розв'язків *Y*.

Крок 3. Перевіряємо виконання нерівності $y^i \geq y^j$, якщо воно виконується, то перейти до кроку 4, інакше перейти до кроку 5.

Крок 4. Видаляємо з поточної множини векторів $P(Y)$ вектор y^j, переходимо до кроку 5.

Крок 5. Перевіряємо виконання нерівності $j < i$, якщо вона виконується, то покладаємо $j = j + 1$ і переходимо до Кроку 7. У іншому випадку необхідно перейти до кроку 8.

Крок 7. Перевіряємо справедливість нерівності $y^i \geq y^j$, якщо вона виконується, переходимо до кроку 7, інакше повертаємося до кроку 5.

Крок 7. Видаляємо з поточної множини векторів $P(Y)$ вектор y^i і переходимо до кроку 8.

Крок 8. Перевіряємо виконання нерівності $i < l - 1$, якщо вона виконується, то покладаємо $i = i + 1$, а потім $j = i + 1$ і переходимо до кроку 3, інакше розрахунки закінчуються. Множина Парето-оптимальних векторів побудована повністю.

Суть алгоритму полягає у тому, що шукана множина Парето-оптимальних векторів утворюється послідовним видаленням завчасно відомих неоптимальних векторів.

Алгоритм ускладнюється, якщо область можливих розв'язків визначається не тільки комбінаторними умовами, які описують граф деякої комбінаторної конфігурації, а накладаються ще додаткові обмеження.

Досліджені розв'язки комбінаторної багатокритерійної задачі, їх властивості та підхід до знаходження ефективних оцінок дають можливість розробити загальний підхід до розв'язання багатокритеріальних задач на комбінаторних конфігураціях. Далі розглянемо багатокритеріальну задачу на комбінаторній конфігурації розміщень та застосування до її розв'язання методу направленого структурування.

7.3. Розв'язування екстремальних задач при умові багатокритеріальності за методом направленого структурування

У даному пункті розглядається випадок, коли X – комбінаторна множина розміщень. $F(x) = \left(f_1(x), \ldots, f_i(x), \ldots, f_l(x)\right)$ – векторний критерій, заданий на множині $A(B)$ розміщень, породжуваних деякою скінченною мультимножиною $B=\left\{b_1,b_2,\ldots,b_q\right\}$. Тоді задача має вигляд: $Z\left(F,X\right)$.

Задача може містити також додаткові лінійні обмеження, які утворять опуклу многогранну множину $D\subset R^n$ вигляду:

$$D=\left\{x\in R^n \mid Gx\le H\right\},$$

де $G\in R^{m\times n}$, $H\in R^m$.

Як відомо, розміщенням з q елементів по n називається впорядкований набір з n елементів, який належать мультимножині $B=\left\{b_1,b_2,\ldots,b_q\right\}$.

Варто зазначити, що в цьому випадку поняття «оптимуму» має несуперечливе означення: допустимий розв’язок x^0 оптимальний, якщо він мінімізує (або максимізує) цільову функцію на

$$X : F\left(x^0,b\right)=\min_{x\in X} F\left(x,b\right)$$

або $F\left(x^0,b\right)=F\left(x^0,b\right)=\max_{x\in X} F\left(x,b\right),$ де X містить елементи множини розміщень $A\left(B\right)$.

Кожний елемент множини $A\left(B\right)$ є впорядкованим набором q відповідних дійсних чисел. Не втрачаючи спільності, упорядкуємо елементи мультимножини B по неспаданню таким способом: $b_1\le b_2\le\ldots\le b_q$. Тоді опуклою оболонкою загальної множини розміщень $A\left(B\right)$ є загальний многогранник розміщень $M=conv\,A\left(B\right)$ [87], який запишемо в наступному вигляді:

$$M=\left\{x\in R^n \left| \sum_{j=1}^{|\omega|} b_j \le \sum_{i\in\omega} x_i \le \sum_{j=1}^{|\omega|} b_{q-j+1} \quad \forall\omega\subset N_n=\left\{1,\ldots,n\right\}\right.\right\},$$

де $A\left(B\right)=vert\,M$.

Згідно [87] $x \in M = conv\, A(B)$ є вершиною многогранника розміщень M тоді й тільки, коли він являє собою перестановку чисел $b_1, \dots, b_s, b_{q-r+1}, \dots, b_q$, де $0 \le s \le q,\ 0 \le r \le n,\ s + r = n$.

Враховуючи що, для переставного многогранника можна побудувати граф, за допомогою якого можна розглядати зміну значення цільової функції в точках – вершинах многогранника розміщень. Скористаємося наступною теоремою.

Многогранник розміщень $M = conv\, A(B) = M_q^n(B)$ при $n < q$ і будь-якому векторі $B = (b_1, b_2, \dots, b_q)$ комбінаторно еквівалентний переставному многограннику $M_n(B)$ розмірності n.

З огляду на зв’язок між перестановками й розміщеннями, для елементів множини розміщень можна побудувати подібний граф многогранника розміщень $M_q^n(B)$. Далі $M_q^n(B) = M$.

Нехай існує граф $G(B)$ комбінаторної конфігурації розміщень M.

Розглянемо приклад для розміщення з 4 елементів по 3. Опишемо побудову графа $G(B)$ для розміщення з 4 елементів по 3. Для зручності подальшого викладу і розуміння, елементи множини розміщення – точки пронумеруємо від 1 до 24 тому що їх буде саме 24, і будемо їх називати вершинами $p_i, i \in N_{24}$ графа $G(B)$, які будуть розміщатися на чотирьох підграфах $G_1(B),\ G_2(B),\ G_3(B),\ G_4(B)$, залежно від вибору елементів із множини B. Тоді для підграфів графа $G(B)$ виконуються наступні умови $X_1 \subseteq X,\ X_2 \subseteq X,\ X_3 \subseteq X,\ X_4 \subseteq X$, де $X,\ X_i$, $i \in N_4$ – множина вершин; $U_1 \subseteq U,\ U_2 \subseteq U,\ U_3 \subseteq U,\ U_4 \subseteq U$, де $U, U_i, i \in N_4$ – множина ребер. У підграфі $G_1(B)$, що розташований у верхній частині графа $G(B)$ вершини утворені з максимальних елементів мультимножини $B = \{b_1, b_2, \dots, b_q\}$, тобто (2, 3, 4), які утворюють елементи множини розміщень. Наступний підграф $G_2(B)$ можна побудувати шляхом заміни мінімального елемента на ще більш мінімальний, тобто вибирається підмножина (1, 3, 4) і генеруються вершини. Аналогічним чином будуються підграфи: $G_3(B)$, вершини яких утворюються шляхом генерування елементів (1, 2, 4) і $G_4(B)$ – вершини утворюються з (1, 2, 3). Далі розглянемо будь-який частковий критерій $f(x) = f_i(x)$, $i \in N_l$ вектор-функції $F(x)$.

Вектор коефіцієнтів функції $f(x) = \sum_{j=1}^{n} \langle c_j, x \rangle$ позначимо $\vec{c} = (c_j)$, $j \in N_n$, де $c_1 \le c_2 \le \ldots \le c_n$, $i \in N_n$ для функції. Тоді значення функції $f(x)$ в довільній точці $p_i (1 \le i \le 24)$ визначається як скалярний добуток $f(p_j) = (p_j, c_j)$.

Далі розглянемо побудову вершин у підграфах. Слід зазначити, що граф $G(B)$ можна побудувати індуктивним способом, починаючи із двох перших елементів розміщення, аналогічно графу многогранника перестановок [60, 61]. Розглянемо одну цікаву властивість у вигляді леми, що будемо використовуватися для побудови графа.

Лема 7.1. Із двох суміжних вершин p_i, p_j графа розміщень M функція $f(x)$ приймає не менше (більше) значення для тієї вершини, у якій максимальний з двох елементів, що різняться, перебуває праворуч, за умови, що коефіцієнти цільової $f(x) = \sum_{j=1}^{n} \langle c_j, x \rangle$ функції впорядковані таким чином $c_1 \le c_2 \le \ldots \le c_n$ $i \in N_n$.

Розглянемо вершини p_1 й p_2, які розміщені на деякому підграфі $G_1(B)$ графа $G(B)$. Нехай маємо вершину $p_1 = (2,3,4) \in G_1(B)$, тоді p_2 утворена із p_1 шляхом транспозиції елементів 1 і 2. Відповідно до леми 5.3 одержимо співвідношення значень цільової функції в цих точках: $f(p_1) \ge f(p_2)$. Якщо тепер у вершинах p_1, p_2 поміняти місцями елементи 2 і 3, то одержимо вершини p_3 й p_5, для яких виконується співвідношення $f(p_3) \ge f(p_5)$. Крім того, по лемі 7.1 одержуємо $f(p_1) \geq f(p_3)$ й $f(p_2) \geq f(p_5)$. Аналогічно, шляхом транспозиції елементів 1 і 3 одержимо з p_2 вершину p_4, а з p_4 транспозицією елементів 2, 3 вершину p_6. У результаті цих дій одержимо повний підграф $G_1(B)$, що містить всі вершини многогранника розміщень *M*, отримані з елементів $(2,3,4)$. Очевидно, що в підграфі $G_1(B)$ функція $f(x)$ приймає максимальне значення у вершині p_1 й мінімальне – у вершині p_6 (при впорядкуванні елементів множини розміщень і коефіцієнтів цільової функції за зростанням).

Шлях (маршрут) на кожному підграфі графа $G(B)$ визначається послідовністю вершин і ребер для першого підграфа $G_1(B)$ відповідно:

$$p_1 u_1 p_2 u_2 p_3 u_3 p_4 u_4 p_5 u_5 p_6 u_6,$$

для другого $G_2(B)$ – послідовністю

$$p_7 u_7 p_8 u_8 p_9 u_9 p_{10} u_{10} p_{11} u_{11} p_{12} u_{12},$$

для третього $G_3(B)$ – послідовністю

$$p_{13} u_{13} p_{14} u_{14} p_{15} u_{15} p_{16} u_{16} p_{17} u_{17} p_{18} u_{18},$$

для четвертого $G_4(B)$ –

$$p_{19} u_{19} p_{20} u_{20} p_{21} u_{21} p_{22} u_{22} p_{23} u_{23} p_{24} u_{24},$$

де $p_i \in vert\,M, u_i \in U, i \in N_{24}$.

Ребро u_i з’єднує вершину p_i з вершиною p_{i+1}, тобто виконується відношення інцидентності $\Phi(p_i, u_i, p_{i+1})$. У кожному підграфі $G_1(B)$, $G_2(B)$, $G_3(B)$, $G_4(B)$ можна визначити гамільтонів цикл, що містить всі вершини підграфа.

Для подальшого викладу матеріалу розглянемо деякі цікаві властивості описаного графа $G(B)$ розміщень згідно (рис. 7.2).

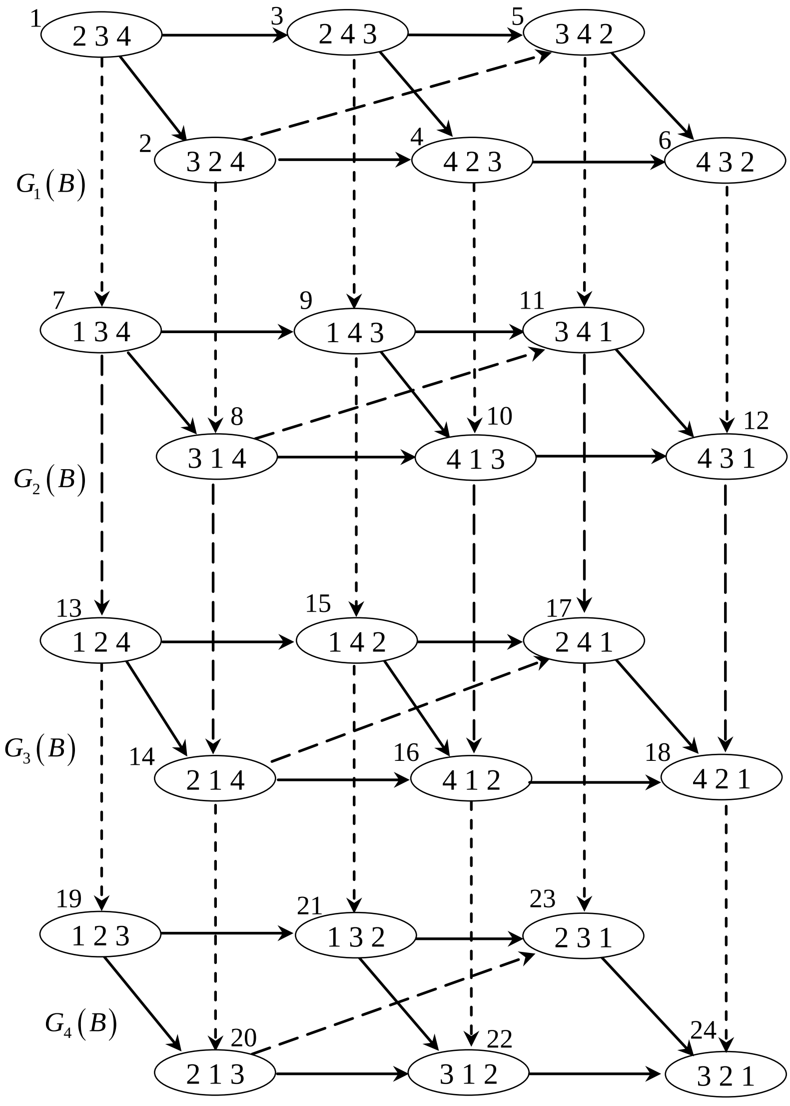


Рис. 7.2. Представлення графа послідовності розміщень

Говорячи про яку-небудь задачу на графі $G(B)$, її допустимий розв'язок позначимо через *x*, маючи на увазі, що $x = (X_x, U_x)$ – це підграф графа $G(B)$ з множиною вершин $X_x \subseteq X$, де X – множина вершин графа $G(B)$ і множиною ребер $U_x \subseteq U$ – множиною всіх допустимих розв'язків цієї задачі, який задовольняє певним умовам. Для розглянутої задачі множина всіх допустимих розв'язків $X = vertM$.

Оскільки побудова даного графу подібна графу перестановок, що описаний у роботі [60, 61] та другому розділі, то доцільно стверджувати, що граф $G(B)$ складається з підграфів, які є скінченними й ізоморфні між собою (рис. 7.2).

Граф $G(B)$ можна розглядати як скінченний граф, що є об'єднанням чотирьох підграфів $G_1(B) = (X_1, U_1)$, $G_2(B) = (X_2, U_2)$, $G_3(B) = (X_3, U_3)$, $G_4(B) = (X_4, U_4)$, де $X_1, X_2, X_3, X_4 \in vert\, M$ й для $G = (X, U)$, $vert\, M = X = X_1 \cup X_2 \cup X_3 \cup X_4$, $U = U_1 \cup U_2$.

Оскільки $G_1(B) = (X_1, U_1)$, $G_2(B) = (X_2, U_2)$, $G_3(B) = (X_3, U_3)$, $G_4(B) = (X_4, U_4)$ графи попарно ізоморфні, тому що між їх вершинами й ребрами існує взаємнооднозначна відповідність, то графи можна однаково зображувати графічно й відрізнятися вони будуть тільки мітками вершин, що й позначено на рис. 7.2.

Граф комбінаторної конфігурації розміщень орієнтований, як це представлено на рис. 7.2 згідно значень лінійної цільової функції

Для цього розглянемо шістки точок – вершини підграфів графа $G(B)$, які можна представити на площині у вигляді плоского (планарного) графа, так само як і для графа перестановок [60]. Причому для кожної шістки вершин можна побудувати гамільтонів шлях. Для довільних *n* граф $G(B)$ розкладається на підграфи, де найменший складається із множини вершин і дуг, що з'єднує ці вершини, та знову приводить до розгляду підграфів вигляду $G_1(B) = (X_1, U_1)$, $G_2(B) = (X_2, U_2)$, $G_3(B) = (X_3, U_3)$, $G_4(B) = (X_4, U_4)$.

З огляду на властивості графа $G(B)$ розглянемо алгоритм розв'язання задачі $Z(F,X)$ на графі розміщень. В алгоритмі використовується метод розв'язування підзадачі, що полягає в знаходженні множини точок – вершин графа за екстремальними значеннях цільових обмежуючих функцій $G_1(x)=\sum_{i=1}^{m} g_{ij}x_i = h_i$, $i \in N_m$, $j \in N_k$, які визначають додаткові обмеження на основі методу ділення відрізка навпіл (методу дихотомії) використовуючи властивості графа $G(B)$.

Алгоритм розв'язання векторної задачі на розміщеннях

Нехай $G(B)=(X,U)$ є граф, задана також множина підграфів $\{G_1,\ldots,G_q\}$. Допустимий розв'язок x визначається як підграф $x=(X_x,U_x), X_x \subseteq X, U_x \subseteq U$, у якому кожний компонент зв'язності ізоморфний графу $G(B)$, де X – множина допустимих розв'язків на графі. З огляду на багатокритеріальність заданої задачі, зазначимо що її необхідно звести до однокритеріальної задачі.

Для цього застосуємо відомий алгоритм лінійної згортки. Ці алгоритми базуються на наступному відомому факті: при додатньо-визначеній вектор-цільової функції елемент $x \in X$, максимізуючий лінійну згортку $F^{\lambda}(x)$, є Парето-оптимальним. Далі загальна ідея запропонованого методу розв'язання задачі $Z(F,X)$ полягає в послідовному розгляді підзадач, кожна з яких містить функції з вектор-функції й функції-обмеження.

Алгоритм

Початковий крок. Вважаємо $s=0$. Зведемо багатокритеріальну задачу $Z(F,G)$ на графі $G(B)$ конфігурації розміщень до однокритеріальної за допомогою лінійною згортки: задаємо вагові невід'ємні коефіцієнти λ_j, $j \in N_l$, які визначають ступінь важливості кожного критерію, і максимізуємо лінійну комбінацію цільових функцій, тобто розв'язуємо задачу

$$Z\left(f, G^s\right),$$

де $f(x) = \sum_{i=1}^{l} \lambda_i \langle c_i, x\rangle$, $\lambda_i \geq 0$, $i \in N_l$, $\sum_{i=1}^{l} \lambda_i = 1$, $G^s = R^n$.

У випадку, якщо який-небудь із коефіцієнтів $\lambda_i = 1$, а всі інші $\lambda_j = 0$, $i \neq j$, $i, j \in N_l$, то розглядається однокритеріальна задача з i-ю цільовою функцією.

Задаємо елементи множини $B = \left(b_1, b_2, \ldots, b_q\right)$, які впорядковані $b_1 \leq b_2 \leq \ldots \leq b_q$ задаємо коефіцієнти цільової функції $c_1, c_2, \ldots, c_n$.

Основна частина

Крок 1. Упорядкуємо коефіцієнти $\lambda_i \cdot c_i$, $i \in N_l$, $\lambda_1 c_1 \leq \lambda_2 c_2 \leq ... \leq \lambda_n c_n$, $i \in N_n$ цільової функції $f(x) = \sum_{i=1}^{l} \lambda_i \langle c_i, x\rangle$.

Крок 2. Визначаємо мінімальні й максимальні значення цільової функції $f(x) = \sum_{i=1}^{l} \lambda_i \langle c_i, x\rangle$, обчислюємо значення $f\left(x^*_{min}\right)$, $f\left(x^*_{max}\right)$ на загальному графі многогранника $G(B)$ й на кожному з підграфів $G_1(B), \ldots, G_n(B)$.

Крок 3. $k = 0$. Вибираємо одне з додаткових обмежень $h_i = \sum_{i=1}^{m} g_{ij} x_i$, $i \in N_m, j \in N_n$ присвоюємо $k = k + 1$, якщо $k = m$, тобто всі обмеження обрані, то на крок 10. Інакше, задаємо коефіцієнти додаткового обмеження k: g_{ij}, $i \in N_m$, $j \in N_k$,

$k = k+1,\ i := k-i$. Будуємо решітки для перетворення індексів коефіцієнтів (упорядковуємо коефіцієнти):

$$U_k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} \Rightarrow U'_k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ \alpha'_1 & \alpha'_2 & \alpha'_3 & \alpha'_4 \end{pmatrix}.$$

Крок 4. Перетворюємо додаткове обмеження в обмеження вигляду: $\tilde{g}_{ij}(x) = U_i g_{ij}(x) \geq h_i$, у якому коефіцієнти впорядковані по зростанню.

Крок 5. Визначаємо $\tilde{g}_k\left(x^*_{max}\right) = max,\ \tilde{g}_k\left(x^*_{min}\right) = min$ на графі $G(B)$.

Крок 6. Знаходимо значення функції додаткового обмеження $\tilde{g}_{ij}(x)$ в лівих крайніх точках $p_{i_{left}}$, які визначають max значень функції $\tilde{g}_k(x)$ на кожному підграфі $G_1(B),\ldots,G_n(B)$ графа, представивши граф в вигляді структурної схеми.

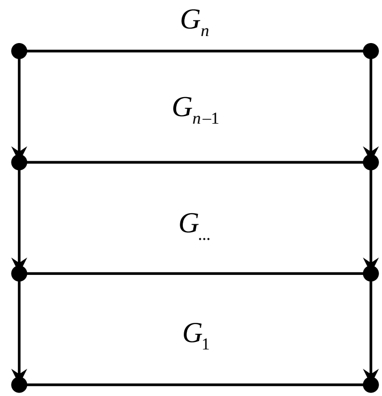


Рис. 7.3. Структурна схема графа $G(P_n)$

Крок 7. Порівнюємо виконання обмежень $\tilde{g}_q(x) \geq b_q$, якщо умова виконується, запам'ятовуємо $p_{i_{left}}$ і переходимо на наступний крок 8. Інакше на крок 9.

Крок 8. Знаходимо значення функції – додаткового обмеження в точках $p_{i_{rite}}$ (точки, які визначають min значення $\tilde{g}_k(x)$ на кожному підграфі). Ділимо відрізок, що визначається точками $p_{i_{rite}} \le g_q(x) \le p_{i_{left}}$ на кроці 6, 7 навпіл і одержуємо точку $\bar{x}^* = \dfrac{\left(p_{i_{left}} - p_{i_{rite}}\right)}{2}$. Переходимо на наступний крок.

Крок 9. Перевіряємо виконання перетвореного додаткового обмеження $g_k(x) \ge b_k$, підставивши значення точки $\bar{x}^*$ із множини розміщень $A(B)$. Якщо нерівність виконується, то запам'ятовуємо потрібний відрізок $\left[\bar{x}^*, p^*_{min}\right]$ або $\left[p^*_{max}, \bar{x}^*\right]$. Перевіряємо виконання умови $k = m,$, якщо не виконується, то переходимо на крок 3. Інакше на наступний крок.

Крок 10. Для всіх додаткових обмежень шукаємо підграф, що визначається множиною вершин і обчислюємо на ньому min або max значень цільової функції $f(x)$. Задача розв'язана, якщо значення цільової функції знаходяться в точках на перетині і визначають об'єднання вершин підграфів $G_1(B),\ldots,G_n(B)$. Інакше – задача нерозв'язна.

Таким чином, результатом роботи алгоритму є підграф $x^0 = (X,U,\Phi)$ вихідного графа $G(B) = (X,U,\Phi)$, з якого вибираються Парето-оптимальні розв'язки, такі як x^0, що задовольняють означенню Парето-оптимального розв'язку задачі $Z(F,X)$. Оскільки розв'язки задачі шукається на скінченній дискретній множині розміщень, то можна гарантувати знаходження хоча б одного Парето-оптимального розв'язку $\tilde{x}$ задачі $Z(F,X)$ з вектор-цільовою функцією, а відповідно, застосування алгоритму до даного графа $G(B)$.

7.4. Розв'язання багатокритеріальних задач оптимізації на полікомбінаторних множинах

Систематичне вивчення властивостей комбінаторних множин та їх дослідження описані в багатьох роботах. Поряд з добре відомими комбінаторними конфігураціями перестановок, розміщень, сполучень, розбиттів виділяються більш складні структури – полікомбінаторні структури [98, 240]. Підвищений інтерес до полікомбінаторних конфігурацій обумовлений дослідженнями останніх років в області комп'ютерних технологій при створенні сучасних алгоритмів і програм для розв'язування оптимізаційних задач. Слід зазначити, що задачі комбінаторної оптимізації на полікомбінаторних множинах невід'ємно пов'язані з комбінаторними многогранниками, що є опуклими оболонками таких множин, і їх властивостями та описані в [99, 245].

В даному розділі продовжується дослідження багатокритеріальних задач на полікомбінаторних множинах перестановок, розміщень, відображені в роботах [99, 245]. На підставі встановленого взаємозв'язку між багатокритеріальними задачами на комбінаторних множинах і оптимізаційними задачами на неперервній допустимій множині, встановлено деякі структурні властивості допустимої області, множин різних видів ефективних розв'язків, а також сформульовано і доведено ряд теорем про умови оптимальності ефективних розв'язків розглянутих задач [206]. Для векторних задач комбінаторного типу на полірозміщеннях запропонований один з можливих підходів до їх розв'язання.

Розглянемо основні поняття та означення, необхідні для постановки задачі та викладу основних результатів даної задачі [99, 245]. Зазначимо, що поняття вибірки, мультимножини означено в попередніх розділах.

Представимо множину N_q у вигляді впорядкованого розбиття на s, де $s<q$, непустих попарно непересічних підмножин $N_1,\ldots,N_s$, тобто для них виконуються умови: $N_i \cap N_j = \varnothing$, $N_i \neq \varnothing$, $N_j \neq \varnothing$, $\forall i,j \in N_s$, а так само впорядковане розбиття числа k на s доданки $k_1,k_2,\ldots,k_s$, що задовольняє умова $1 \le k_i \le q_i$, $\forall i \in N_s, |N_i| = q_i$. Очевидно, що $q_1+q_2+...+q_s=q$, $k_1+k_2+...+k_s=k$.

Позначимо H – множину елементів вигляду:

$$h=\left(h(1),\ldots,h(k)\right)=\left(h^1,\ldots,h^s\right),$$

де $h(j) \in N_n$, $j \in N_k$, а h^i – довільна перестановка елементів множини $J_i \; \forall i \in N_s$.

Нехай підмультимножина A^i мультимножини A, складається з тих елементів A, номера яких належать множині N_i: $A^i = \left\{ a_1^i, \ldots, a_{k_i}^i \right\}$, $|N_i| = k_i$.

Як відомо [99, 245], множину

$$A_{qk}^{ns}(A,H) = \left\{ \left(a_{h(1)},\ldots,a_{h(k)} \right) \middle| a_{h(i)} \in A \; \forall i \in N_n, \forall h \in H \right\} \subset R^k$$

називають загальною множиною поліровзміщень, зазначених $A_{qk}^{ns}(A,H) = A_{qk}^{ns}$.

Опуклою оболонкою множини $A_{qk}^{ns}(A,H)$ поліровзміщень є многогранник $M_{qk}^{ns}(A,H)$ поліровзміщень, $M_{qk}^{ns}(A,H) = = conv\, A_{qk}^{ns}(A,H)$, множина вершин якого є підмножиною множини поліровзміщень: $vert\, M_{qk}^{ns}(A,H) \subset A_{qk}^{ns}(A,H)$.

Множина $M_{qk}^{ns}(A,H)$ визначається сукупністю всіх розв’язків системи:

$$\left\{ \sum_{j=1}^{n_i} x_j \le \sum_{j=1}^{n_i} a_j^i, i \in N_s, \quad \sum_{j=1}^{m_i} x_{\alpha_j} \ge \sum_{j=1}^{m_i} a_j^i, \right.$$

$m_i \in N_{q_i-1}, \alpha_j \in N_i, \forall i \in N_s$, $\alpha_j \ne \alpha_t, \forall j \ne t, \forall j,t \in N_i$, при умові впорядкування елементів мультимножини A за неспаданням: $a_1 \le \ldots \le a_k$. Очевидно, що це впорядкування зберігається й для кожної підмультимножини A^i $i \in N_s$, з A.

Зрозуміло, що кожний з многогранників $M_{q_i k_i}^{n_i}(A^i)$ являє собою многогранник розміщень, а отже визначає об’єкти конфігурації розміщень.

Оскільки для конфігурації розміщень відомий орієнтований граф по значеннях лінійної функції, то його представлення можна використати і для поліровзміщень, згідно означення добутку многрагранників і відповідної рівності

$$\bigotimes_{i=1}^{s} M_{q_i k_i}^{n_i}\left(A^i\right)=\left\{x \in R^{d_1+\ldots+d_s} \mid x=\left(x_1,\ldots,x_s\right),\ x_i \in M_{q_i k_i}^{n_i}\left(A^i\right)\ \forall i \in N_s\right\},$$

де точка $x \in \bigotimes_{i=1}^{s} M_{q_i k_i}^{n_i}\left(A^i\right)$ задовольняє кожній з s підсистем системи

$$\begin{cases} \sum_{j \in \omega^i} x_j \geq \sum^{\left|\omega^i\right|} a_j^{N_i} \\ \sum_{j \in \omega^i} x_j \leq \sum^{\left|\omega^i\right|} a_{\eta_i - j+1}^{N_i} \end{cases} \quad \forall \omega^i \subset N_i', \forall i \in N_s . \tag{7.5}$$

Отже, можна стверджувати, що якщо a_{h^i} – вершина многогранника $M_{q_i k_i}^{n_i}\left(A^i\right)$, то $a(h)=\bigotimes_{i=1}^{s} a_{h^i}$. Відповідно

$$a(h)=\left(a_{h^1},\ldots,a_{h^s}\right),$$

де $a(h) \in P_{qk}^{ns}\left(A,H\right)$.

Розглянемо структуру безумовної багатокритеріальної задачі на полірозміщеннях. Через N_m, N_s позначаємо множини m та s перших натуральних чисел відповідно, $N_m=\{1,\ldots,m\}$, $N_s=\{1,\ldots,s\}$.

Критерії, що оптимізуються, можна представити набором функцій:

$$f_1(x)=\sum_{j=1}^{k} c_j^1 x_j \to min;$$

$$f_2(x)=\sum_{j=1}^{k} c_j^2 x_j \to min;$$

$$\ldots$$

$$f_s(x)=\sum_{j=1}^{k} c_j^s x_j \to min;$$

$$f_{s+1}(x)=\sum_{j=1}^{k} c_j^{s+1} x_j \to max;$$

$$\ldots$$

$$f_m(x)=\sum_{j=1}^{k} c_j^{m} x_j \to max. \qquad (7.6)$$

Тобто з m функцій s мінімізуються, а $m-s$ навпаки максимізуються. Але у практичному застосування часто виникає потреба у зменшенні одних критеріїв та збільшенні інших. Частинним випадком цієї ситуації буде випадок, коли всі функції максимізуються або мінімізуються.

Набір функцій (7.6) як відомо можна представити у вигляді вектор-функції:

$$\vec{F}\left(-f_1,\ldots,-f_s, f_{s+1},\ldots, f_m\right) \qquad (7.7)$$

максимум якої нам необхідно знайти.

Умова належності розв'язків множині полірозміщень може виникати з додаткових умов, що накладаються на змінні в самій постановці задачі. У побудованій математичній моделі на розв'язок накладається умова належності множині полірозміщень у вигляді:

$$x=\left(x_1,\ldots,x_k\right)\in A_{qk}^{ns}\left(A,H\right). \qquad (7.8)$$

З урахуванням усіх вище означених умов, задачу можна сформулювати наступним чином: знайти множину значень (7.8), що є оптимальними для функції (7.7).

Таку задачу назвемо комбінаторною багатокритеріальною безумовною задачею на множині полірозміщень. Якщо на множину допустимих розв'язків накладаються додаткові умови вигляду:

$$a_{ij}x_j \le b_j,$$

де $i\in N_m, j\in N_k,$ (7.9)

то задача вигляду (7.7)–(7.9) є комбінаторною багатокритеріальною з додатковими обмеженнями.

Підхід до розв’язування комбінаторних багатокритеріальних задач на полірозміщеннях

При розв’язуванні багатокритеріальних комбінаторних задач постає питання визначення ефективного розв’язку, що пов’язане з порівнянням альтернатив на множині цільових функцій. Слід зазначити, що такий розв’язок може виявитись не оптимальним для жодної з цільових функцій, проте, він є найкращим компромісним розв’язком з урахуванням усіх цільових функцій (критеріїв) одночасно.

Визначення 7.5 [171, 172]. Альтернатива x_0 має назву ефективної, якщо на множині допустимих альтернатив *X*, що визначається умовами (7.8), (7.9), не існує такої альтернативи $\overline{x}$, для якої виконувались би нерівності:

$$f_i\left(\overline{x}\right) \geq f_i\left(x_0\right), \forall i \in I_1$$
$$f_i\left(\overline{x}\right) \leq f_i\left(x_0\right), \forall i \in I_2 \qquad (7.10)$$

і хоча б одна з них була строгою.

Це означає, що жодна з допустимих альтернатив не може покращити значення деякої цільової функції, не погіршуючи при цьому хоча б одну з цільових функцій, що залишились. Ефективну альтернативу називають також оптимальною по Парето. Усі такі альтернативи складають множину Парето-оптимальних розв’язків $P_F\left(X\right)$ комбінаторної задачі на полірозміщеннях. Крім Парето-оптимальних (ефективних) розв’язків визначають також і інші множини розв’язків задач комбінаторної багатокритеріальної оптимізації на полірозміщеннях, такі як $Sl_F\left(X\right)$ – множина оптимальних по Слейтеру (слабко ефективних) розв’язків, $Sm_F\left(X\right)$ – множина оптимальних по Смейлу (строго ефективних) розв’язків. Такі множини для багатокрітеріальних задач на перестановках були розглянуті в попередньому пункті.

Вивчення властивостей даних множин є необхідним для дослідження досить важливого та актуального питання стійкості оптимізаційних багатокритеріальних задач на полікомбінаторних множинах та стійкості їх розв'язків. Це пов'язано з тим, що вихідні дані задач, які є математичними моделями різноманітних процесів, в більшості випадків не можуть бути визначені однозначно. Вони задаються з деякою похибкою і залежать від багатьох параметрів, а тому потребують уточнення в процесі розв'язування задачі. Для багатьох задач такі зміни значень критеріїв приводять до суттєвих змін результату, а це в свою чергу до неточності істинного розв'язку. Тому необхідним кроком є дослідження питання стійкості розв'язків задач, виділення класів стійкості та розробка методів, які б дозволяли змінити чисельний розв'язок некоректних задач із непередбаченим впливом збурень у вихідних даних задачі, що докладно пояснюється в роботах [217, 219].

Комбінований метод для розв'язування комбінаторної багатокритеріальної задачі оптимізації на полірозміщеннях

Розглядається метод, що є поєднанням двох методів: методу обмежень [60, 105] та методу комбінаторного відсікання [97, 243].

Як зазначалося, комбінаторні багатокритеріальні задачі є досить актуальними при розв'язанні ряду прикладних задач, але розроблені існуючі методи не повністю адекватно можуть знаходити розв'язок таких задач. Є доцільним розробити новий підхід до їх розв'язання.

Розглянемо метод обмежень [105, 117], адаптований до вищезазначених позначень.

Нехай задана деяка множина цільових функцій $f_i(x)$, де

$$f_i(x) = \sum_{j=1}^{k} c_{ij} x_j, i \in N_m$$

причому s перших функцій треба мінімізувати, а наступні $m-s$ – максимізувати, що записано у вигляді вектор-функції (7.7) На результуючий вектор $X = \{x_j\}, j \in N_k$ накладені обмеження виду (7.9) а також умова належності розв'язку множині полірозміщень (7.8).

Алгоритм розв’язування задачі

1. Умова належності розв’язку множині полірозміщень записується у вигляді системи нерівностей, що описують відповідну комбінаторну множину. Покладаємо значення цілочислової змінної $k = 0$.

2. Об’єднується система (7.5) пункту 1 з системою лінійних обмежень (7.9) задачі.

3. Визначаються для кожної з функцій такі розв’язки, що задовольняють обмеження (7.8) і (7.9), а також мінімізують та максимізують функції, підставивши відповідні значення у кожну з функцій.

4. Застосовуються наступні відображення:

а) для функцій, що мінімізуються:

$$W_i\left(f_i\left(X\right)\right)=\frac{\sum_{j=1}^{k} c_{ij}x_j-\sum_{j=1}^{k} c_{ij}x_j^0}{\sum_{j=1}^{k} c_{ij}x_{i\,max}-\sum_{j=1}^{k} c_{ij}x_j^0},\ \ \forall i\in N_s; \qquad (7.11)$$

б) для функцій, що максимізуються:

$$W_i\left(f_i\left(X\right)\right)=\frac{\sum_{j=1}^{k} c_{ij}x_j^0-\sum_{j=1}^{k} c_{ij}x_j}{\sum_{j=1}^{k} c_{ij}x_j^0-\sum_{j=1}^{k} c_{ij}x_{i\,min}},\ \ \forall i\in N_{m-s}, \qquad (7.12)$$

де x_j^0 – розв’язок, що задовольняє умовам (7.11), (7.12), та оптимізує i-ту цільову функцію, а $x_{max}\left(x_{min}\right)$ – розв’язки, що максимізують (мінімізують) відповідний критерій на допустимій множині розв’язків.

5. Записується наступна задача лінійного програмування:

$$k_0 = x_{n+1} \to min$$

$$
\begin{cases}
\sum_{j=1}^{k} c_{ij} x_j \le \sum_{j=1}^{k} c_{ij} x_j^0 + \dfrac{k_0}{\rho_i}\left(\sum_{j=1}^{k} c_{ij} x_{i\,max} - \sum_{j=1}^{k} c_{ij} x_j^0 \right), \forall i \in N_s \\
\sum_{j=1}^{k} c_{ij} x_j \ge \sum_{j=1}^{k} c_{ij} x_j^0 - \dfrac{k_0}{\rho_i}\left(\sum_{j=1}^{k} c_{ij} x_j^0 - \sum_{j=1}^{k} c_{ij} x_{i\,min} \right), \forall i \in N_{m-s} \\
a_{ij} x_j \le b_j, i \in N_n \\
\sum_{j \in \omega^i} x_j \ge \sum^{|\omega^i|} a_j^{N_i} \\
\sum_{j \in \omega^i} x_j \le \sum^{|\omega^i|} a_{q_i - j + 1}^{N_i}
\end{cases}
\qquad (7.13)
$$

Розв’язуємо її двоїстим симплекс-методом.

Зауваження до пункту 5. Компромісним розв’язком даної багатокритеріальної задачі буде такий ефективний розв’язок x, для якого відносні відхилення однакові та мінімальні, тобто:

$$
\rho_1 W_1(X) = \rho_2 W_2(X) = \ldots = \rho_m W_m(X) = k_{0\,min}. \qquad (7.14)
$$

6. Задача пункту 5 перетворюється до вигляду:
мінімізувати $k_0 = x_{n+1}$ при обмеженнях

$$
\sum_{j=1}^{n} d_{1j} x_j + d_{1n+1} x_{n+1} + d_1 \ge 0;
$$

$$
\sum_{j=1}^{n} d_{ij} x_j + d_{in+1} x_{n+1} + d_i \ge 0;
$$

$$
\ldots
$$

$$
\sum_{j=1}^{n} d_{mj} x_j + d_{mn+1} x_{n+1} + d_m \ge 0;
$$

$$\left.\begin{array}{l}\sum_{j=1}^{n} a_{ij} x_j - b_1 \le 0 \\ \sum_{j=1}^{n} a_{kj} x_j - b_k \le 0\end{array}\right\}, \ \ x_j \ge 0, j \in N_n,$$

де коефіцієнти визначаються наступним чином:

$$d_{ij} = \begin{cases} -\rho_i c_{ij}, \forall j \in N_n; i \in N_s \\ \rho_i c_{ij}, \forall j \in N_n; i \in N_{m-s} \end{cases}$$

$$d_{i,n+1} = \begin{cases} \sum_{j=1}^{n} c_{ij} \left(x_{ij}^{(0)} - x_{ij\,min} \right), i \in N_s \\ \sum_{j=1}^{n} c_{ij} \left(x_{ij\,max} - x_{ij}^{(o)} \right), i \in N_{m-s} \end{cases}$$

$$d_i = \begin{cases} \rho_i \sum_{j=1}^{n} c_{ij} x_{ij}^{(0)}, i \in N_s \\ -\rho_i \sum_{j=1}^{n} c_{ij} x_{ij}^{(o)}, i \in N_{m-s}. \end{cases}$$

Покладаємо $\rho_i = \dfrac{1}{m}, \forall i \in N_m$.

7. Перевіряємо належність вектора-розв’язку $x = \left(x_1, \ldots, x_n\right)$ полікомбінаторній множині (7.8). Якщо $x = \left(x_1, \ldots, x_n\right) \in A_{qk}^{ns}\left(A, H\right)$, то розв’язок знайдено і алгоритм завершує свою роботу, інакше переходимо до пункту 8.

8. Перевіряємо $k > 1$. Якщо «так», то робимо перехід на крок 10. Інакше – перехід на крок 9.

9. Збільшуємо k на одиницю: $k = k + 1$. Додаємо до системи обмежень сформовану нерівність-відсікання:

$$\frac{x_{i_1}}{\Theta_{i_1}}+\frac{x_{i_2}}{\Theta_{i_2}}+...+\frac{x_{i_\gamma}}{\Theta_{i_\gamma}}\geq 1 \qquad (7.15)$$

у вигляді рівності

$$-\frac{x_{i_1}}{\Theta_{i_1}}-\frac{x_{i_2}}{\Theta_{i_2}}-...-\frac{x_{i_\gamma}}{\Theta_{i_\gamma}}+x_{n+q}=-1, \qquad (7.16)$$

ввівши допоміжну змінну $x_{n+q}\geq 0$. У формулах (7.15), (7.16) $i_1,\ldots,i_\gamma$ – номери небазисних змінних в останній точці x^*, γ – їх кількість, а $\Theta_{ij}\forall j\in N_\gamma$ знаходиться так:

$$\Theta_i=\min_{j:a_{ij}>0}\frac{b_i}{a_{ij}}=\frac{b_t}{a_{tj}}. \qquad (7.17).$$

Переходимо на крок 5.

10. Перевіряємо: $\Theta_{n+q-1}=0$ (Θ_{n+q-1} обчислюється за формулою (7.17). Якщо «ні», переходимо на крок 9. Якщо «так», то в останній приєднаній до системи рівності (7.16) замінити введену там допоміжну змінну на 0. Переходимо на крок 5 алгоритму.

Приклад розв’язання задачі. Нехай задана мультимножина $A=\{1,2,3,3,4\}$, що містить 5 елементів. Отже $N_5=\{1,2,3,4,5\}$. Нехай задано $s=2$, виберемо розбиття N_5 на множини $N_1=\{1,3,5\};N_2=\{2,4\}$. Нехай задано $k=2$ та вибрані $k_1=1$, $k_2=1$. Тоді множина H утворюється у вигляді: $H=\{(1,2);(1,4);(3,2);(3,4);(5,2);(5,4)\}$. А отже, множина полірозміщень матиме такий вигляд:

$$A_{54}^{22}(A,H)=\{(1,2);(1,3);(3,2);(3,3);(4,2);(4,3)\}.$$

Математична постановка: знайти множину значень $x\in A_{54}^{22}(A,H)$, що є оптимальними для функцій

$$f_1(x) = 2x_1 + 3x_2;$$

$$f_2(x) = -x_1 - 2x_2.$$

Розв’язання. Запишемо многогранник полірозміщень (7.30) у вигляді системи обмежень:

$$\begin{cases} x_1 \geq 1 \\ -x_1 \geq -4 \\ x_2 \geq 2 \\ -x_2 \geq -3 \\ x_1 + x_2 \geq 3 \\ -x_1 - x_2 \geq -7 \end{cases}.$$

Скориставшись формулами (7.11), (7.12), перетворимо функції. Для цього проаналізуємо кожну з функцій на найбільше та найменше значення:

x_1	x_2	F_1	F_2
1	2	8	–5
1	3	11	–7
3	2	12	–7
3	3	15	–9
4	2	14	–8
4	3	17	–10

Отже, отримаємо наступні функції:

$$W_1 = \frac{1}{2}\frac{17 - 2x_1 - 3x_2}{17 - 8} = \frac{17 - 2x_1 - 3x_2}{18} \leq x_3$$

$$W_2 = \frac{1}{2}\frac{-x_1 - 2x_2 + 10}{-5 + 10} = \frac{-x_1 - 2x_2 + 10}{10} \leq x_3.$$

Згідно методу обмежень, необхідно розв’язати таку задачу: мінімізувати x_3 при умовах:

$$\begin{cases} 2x_1 + 3x_2 + 18x_3 \geq 17 \\ x_1 + 2x_2 + 10x_3 \geq 10 \\ x_1 \geq 1 \\ -x_1 \geq -4 \\ x_2 \geq 2 \\ -x_2 \geq -3 \\ x_1 + x_2 \geq 3 \\ -x_1 - x_2 \geq -7 \end{cases}$$

При переході до двоїстої задачі отримаємо:

$$17y_1 + 10y_2 + y_3 - 4y_4 + 2y_5 - 3y_6 + 3y_7 - 7y_8 \to max$$

$$\begin{cases} 2y_1 + y_2 + y_3 - y_4 + y_7 - y_8 \leq 0 \\ 3y_1 + 2y_2 + y_5 - y_6 + y_7 - y_8 \leq 0 \\ 18y_1 + 10y_2 \leq 1 \end{cases}$$

Розв’язуючи дану задачу симплекс-методом, отримали у результаті наступні значення $x_1 = 4,$ $x_2 = 3,$ $x_3 = 0,$ отже $x = (4,3) \in A_{54}^{22}(A,H)$ – шуканий розв’язок.

Висновки до розділу

Формулюється постановка задачі комбінаторної оптимізації, що об’єднує проблему багатокритеріальності та комбінаторні властивості розв’язків. Досліджено складні задачі на комбінаторної множині розміщень із багатьма критеріями. Розглянуто деякі властивості допустимої області комбінаторної задачі, що має специфічні вхідні дані. Побудовано й обґрунтований застосування методу направленого структурування до розв’язування екстремальних задач з лінійними функціями цілі на комбінаторній множині розміщень. Отримано розв’язок багатокритеріальної задачі комбінаторної оптимізації на полірозміщеннях з багатьма критеріями та додатковими лінійними обмежень за допомогою комбінованого методу.

Розглянута модель багатокритеріальної задачі з урахуванням комбінаторних властивостей області допустимих розв'язків, яка може бути успішно застосована при розв'язанні різних практичних задач. Встановлено взаємозв'язок між багатокритеріальною задачею на комбінаторних конфігураціях та багатокритеріальною задачею на графі перестановок, розміщень, запропонований загальний метод направленого структурування, що об'єднує засоби комбінаторного аналізу та теорії графів і передбачає послідовне виконання трьох стадій: генерування у певній послідовності всіх елементів заданої комбінаторної конфігурації; побудову на її елементах орієнтованого графа, де дуга відповідає спаданню значень цільової функції; побудову алгоритму розв'язку задачі на частково упорядкованих вершинах графа.

Завдяки частковій упорядкованості часто вдається трудомісткість алгоритмів за даним методом звести до полінома від логарифму всієї кількості комбінаторних конфігурацій.

У результаті проведених досліджень використані структурні властивостей комбінаторних конфігурацій та їх графів, які дають можливість розробляти ефективні алгоритми розв'язання нових класів багатокритеріальих задач на інших комбінаторних конфігураціях.

ПІСЛЯМОВА

Після прочитання цієї монографії у читача повинно виникнути багато запитань до авторів. В цьому останньому слові ми хочемо передбачити їх зміст і заздалегідь на них відповісти. Зрозуміло, що перший і другий розділи не викликають непорозумінь, бо вони носять вступний характер і являються допоміжними до основної теми. Але вже починаючи з третього розділу, матеріал книги викликає деяке відчуття незавершеності і спонукає до діалогу і детальнішого обговорення викладених результатів. В основному мова буде йти про деякі додаткові моменти та невикористані можливості для отримання нових, узагальнених результатів.

В третьому розділі мова йде про методи генерації комбінаторних конфігурацій, серед яких особливу увагу приділено перестановкам. Звичайно, так як кількість перестановок дорівнює $n!$, то будь-який порядок їх переліку можна назвати їх генерацією. Теоретично можливо $(n!)!$ генерацій перестановок з n елементів. В розділі наведено тільки найбільш відомі методи (лексикографічний, антилексикографічний, метод з мінімальною кількістю транспозицій, метод з транспозиціями сусідніх елементів) та додано власний метод переміщення максимального елемента. Граф перестановок побудовано лише для антилексикографічного методу як оптимального для лінійної цільової функції. Проте цікаво було б побудувати такий граф для методу з транспозиціями сусідніх елементів, в якому граф представляється у вигляді гамільтонового ланцюга, складеного з n шляхів, на яких функція почергово то спадає, то зростає. Ті ж побажання можна виказати відносно графів інших конфігурацій, які наведені далі. І взагалі, питання про співвідношення методу генерації комбінаторної конфігурації з відповідним графом цієї конфігурації та типом цільової функції має принциповий характер, тому потребує окремого, детальнішого розгляду. Тим більше, що це має пряме відношення до запропонованого методу розв'язування екстремальних задач на комбінаторних конфігураціях, який має назву методу направленого структурування, конспективний виклад якого наведено в кінці даного розділу.

В четвертому розділі розглядається задача оптимізації лінійної функції на перестановках. В процесі розв'язання цієї задачі доводиться розв'язувати декілька важливих підзадач допоміжного характеру. Одна з них така: як за номером перестановки в даному методі генерації встановити саму перестановку. Існує і обернена підзадача: як за виглядом перестановки знайти її порядковий номер у послідовності згенерованих за даним методом перестановок. Ці підзадачі розв'язані тільки для лексикографічного методу генерації, хоча цікаво розробити алгоритми їх розв'язання і для інших важливих методів. В зв'язку з цим напрошується питання: чи можливо побудувати алгоритм обчислення значення функції за номером перестановки, який вона має в даному способі генерації. Особливо треба виокремити підзадачу про локалізацію значення функції,суть якої полягає в знаходженні такої перестановки (або декількох), на якій функція приймає певне задане значення. В розділі пропонуються два методи розв'язання цієї проблеми для антилексикографічного методу генерації – горизонтальний та координатний. В дечому вони мають схожість, тому було б бажано провести порівняльний аналіз та визначити їх складність. Для невеликих значень кількості вершин графа конфігурацій розроблено прийоми знаходження гамільтонового шляху, на якому лінійна функція монотонно спадає. Було б доречно розробити це питання більш детально та для більших розмірів графів. Знову ж таки недостатньо, чисто конспективно висвітлені проблеми, які виникають для перестановок з повтореннями елементів. Але особливо відчувається повна відсутність розв'язання (або навіть постановки) тих же підзадач для інших типів комбінаторних конфігурацій.

В п'ятому розділі розглядається екстремальна задача на перестановках з дробово-лінійною цільовою функцією, тобто функцією, заданою як відношення двох лінійних функцій. Серед всіх таких функцій для початку вибрано найпростішу, де лінійні функції складають арифметичні прогресії. Доведено, що для таких функцій граф перестановок ізоморфний графу перестановок з лінійною цільовою функцією. Це означає, що граф перестановок представляє ієрархічну структуру зі спадаючою функцією відносно підграфів. У авторів є впевненість, що для довільної дробово-лінійної функції структура графів змінюється таким чином, що можна досить нескладно виявити підграфи, на яких досягаються локальні екстремуми цільової функції. Це потребує окремого дослідження, для якого не знайшлося місця в цьому розділі. І знову, як і в попередніх розділах, треба відзначити недостатню увагу, приділену іншим комбінаторним конфігураціям.

В шостому розділі розглядається загальна схема застосування методу направленого структурування до розв’язання стандартних екстремальних задач на комбінаторних конфігураціях, тобто задач з обмеженнями. Цей розділ, на відміну від інших, написано достатньо повно, тобто в ньому метод демонструється не тільки на перестановках, а і на інших комбінаторних конфігураціях, крім того, не тільки для лінійної функції, а і для дробово-лінійної. Це дає змогу будувати подібні схеми для більш загальних випадків. Особливий науковий інтерес серед інших функцій викликають такі як квадратична та інші поліноміальні функції.

Останній, сьомий розділ, присвячений розв’язуванню оптимізаційних багатокритеріальних задач, – можна вважати, не має самостійного значення, бо він є синтезом сучасної теорії багатокритеріальної оптимізації та методів дискретної оптимізації на комбінаторних конфігураціях, викладених в попередніх розділах. Виявилося, що для цих задач, як і для однокритеріальних, можна з успіхом застосовувати ті ж методи побудови графів комбінаторних конфігурацій, локалізації значень функцій на графах, пошуку екстремума на частково упорядкованих множинах тощо.

Підводячи підсумки, можна сказати, що дана книга є, перш за все, попередньою заявкою на дослідження досить об’ємної галузі дискретної математики з багатообіцяючим майбутнім, і цим самим вона повинна привернути увагу не тільки досвідчених вчених, але і молодих початківців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Айгнер М. Комбинаторная теория / М. Айгнер. − М. : Мир, 1982. – 558 с.
2. Акулич И. Л. Математическое программирование в примерах и задачах / И. Л. Акулич. – М. : Высшая школа, 1986. – 319 с.
3. Алексеев В. Б. Использование симметрии при нахождении ширины частично упорядоченного множества / В. Б. Алексеев // Дискретный анализ. – 1974. − Вып. 26. − С. 20–35.
4. Альбертьян М. К. О комбинаторных характеристиках несравнимости в задачах принятия решений / М. К. Альбертьян // Известия АН СССР. – 1974. − № 6. − С. 3–12. – (Серия «Техническая Кибернетика»).
5. Альседоров З. М. Представление и восстановление графов / З. М. Альседоров, Г. А. Донец. – К. : Наук. думка, 1991. – 188 с.
6. Андрейчиков А. В. Анализ, синтез, планирование решений в экономике / А. В. Андрейчиков, О. Н. Андрейчикова. – М. : Финансы и статистика, 2001. – 364 с.
7. Асакер Р. Конечные графы и сети / Р. Асакер, Т. Саати. − М. : Наука, 1974. – 368 с.
8. Бабич М. Д. Вычислительный эксперимент в проблеме оптимизации вычислений / М. Д. Бабич, В. К. Задирака, И. В. Сергиенко // Кибернетика и системный анализ. – 1999. – № 1 – С. 51–63.
9. Базара М. Нелинейное программирование. Теория и алгоритмы / М. Базара, К. Шетти. – М. : Мир, 1982. – 584 с.
10. Бакурова Г. В. Об устойчивости многокритериальных задач на системах подмножеств / Г. В. Бакурова, В. А. Емеличев, В. А. Перепелица // Доклады АН Беларуси. – 1993. – № 11 – С. 80–84.
11. Баранов В. И. Экстремальные комбинаторные задачи и их приложения / В. И. Баранов, Б. С. Стечкин. – М. : Наука, 1989. – 160 с.
12. Басакер Р. Конечные графы и сети / Р. Басакер, Т. Саати. − М. : Наука, 1974. – 368 с.
13. Белкин А. Р. Принятие решений: комбинаторные модели аппроксимации информации / А. Р. Белкин, М. Ш. Левин. – М. : Наука, 1990. – 232 с.

14. Белов Ю. А. Об одном классе специальных перестановочных многогранников / Ю. А. Белов // Моделирование и анализ информационных систем. – 1996. – № 3 – С. 78–84.
15. Березовский Б. А. Бинарные отношения в многокритериальной оптимизации / Б. А. Березовский, В. И. Борденко, В. И. Кемпнер. – М. : Наука, 1981. – 149 с.
16. Берж К. Теория графов и ее применение / К. Берж. − М. : Изд-во иностр. лит., 1962. − 320с.
17. Білоус Н. В. Основи комбінаторного аналізу / Н. В. Білоус, З. В. Дудар, Н. С. Лесна, І. Ю. Шутін. – Х. : ХДТУРЕ, 1999. – 97 с.
18. Бондаренко В. А. Об одном классе многогранников и их использовании в комбинаторной оптимизации / В. А. Бондаренко // Доклады АН России. – 1993. – Т. 328. – № 3. – С. 303–304.
19. Бренстед А. Введение в теорию выпуклых многогранников / А. Бренстед. – М. : Мир, 1988. – 240 с.
20. Бурдюк В. Я. Разрешимые случаи новой комбинаторной задачи оптимизации / В. Я. Бурдюк, В. А. Семенов // Кибернетика и систем. анализ. – 1999. – № 2. – С. 175–178.
21. Бурштейн Ф. В. Многокритериальные задачи принятия решений при неопределенности и риске / Ф. В. Бурштейн, Э. С. Королев // Теоретическая кибернетика. – Тбилиси, 1980. – С. 156–162.
22. Вейль Г. Элементарная теория выпуклых многогранников : матричные игры / Г. Вейль. – М. : Физматгиз, 1961. – С. 254–273.
23. Вентцель Е. С. Исследование операций: задачи, принципы, методология / Е. С. Вентцель. – М. : Наука, 1988. – 288 с.
24. Веселов С. И. Комбинаторно-алгебраические и вероятностные методы и их применение / С. И. Веселов, А. Ю. Чирков. Горький, 1990. С. 107–110.
25. Вилкас Э. Й. Существование эффективно-равновесных точек в задаче векторной оптимизации / Э. И. Вилкас // Литовский мат. сб. – 1968. – Т. 8. – № 1. – С. 41–44.
26. Виноградская Т. М. Среднее значение числа неподчиненных решений в многокритериальных задачах // Известия АН СССР. – 1976. − № 2. − С. 36–38. – (Серия «Техническая кибернетика»).

27. Виноградская Т. М. Точная верхняя оценка числа неподчиненных решений в многокритериальных задачах / Т. М Виноградская, М. Г. Гафт // Автоматика и телемеханика. – 1974. − № 9.− С. 111–118.
28. Волкович В. Л. Об одной общей схеме последовательного анализа и отсеивания вариантов / В. Л. Волкович, А. Ф. Волошин // Кибернетика. – 1978. – № 5. – С. 98–105.
29. Волконский В. А. О множестве эффективных точек в линейных многокритериальных задачах / В. А. Волконский, Г. К. Еганян, А. Б. Поманский // Сибирский математический журнал. − 1983. – Т. 24. – № 2. – С. 9–17.
30. Волошин О. Ф. Теорія прийняття рішень : навч. посіб. / О. Ф. Волошин, С. О. Мащенко. – К. : Видавничо-поліграфічний центр «Київський університет», 2006. – 304 с.
31. Воронин А. Н. Векторная оптимизация динамических систем / А. Н. Воронин. – К. : Техніка, 1999. – 284 с.
32. Вычислительные методы выбора оптимальных проектных решений / под ред. В. С. Михалевича. – К. : Наукова думка, 1977. – 178 с.
33. Гамидов Р. Г. О принятии решения в задачах многокритериальной оптимизации / Р. Г. Гамидов, М. Ш. Фарбер // Известия АН Азербайджанской РСР. – 1978. – № 3. – С. 11–16. – (Серия физико-технических и математических наук).
34. Гаращенко И. В. Приближенный алгоритм решения симметричной задачи коммивояжера / И. В. Гаращенко, А. В. Панишев, Д. Д. Плечистый // Искусственный интеллект. – 2006. − № 3. – С. 317–378.
35. Гасс С. Линейное программирование / С. Гасс. – М. : Физ.-мат. литература, 1961. – 304 с.
36. Гафт М. Г. Принятие решений при многих критериях / М. Г. Гафт. – М. : Знание, 1979. – 178 с.
37. Герасин С. Н. Покрытие множеств и отношение толерантности / С. Н. Герасин, С. В. Яковлев // Кибернетика и системный анализ. − 2008. – № 3. – С. 29–38.
38. Гирлих Э. Условия разрешимости векторных задач с помощью линейной свертки критериев / Э. Гирлих, М. М. Ковалев, М. К. Кравцов // Кибернетика и системный анализ. – 1999. – № 1. – С. 81–95.

39. Глушков В. М. О системной оптимизации / В. М. Глушков // Кибернетика. – 1980. – № 5. – С. 89–90.
40. Глушков В. М. Основы безбумажной информатики / В. М. Глушков. – М.: Наука, 1982. – 552 с.
41. Гнусов Ю. В. Методологические аспекты прогнозирования функционирования социально-экономических систем / Ю. В. Гнусов, В. В. Тулупов, С. В. Яковлев // Збірник наукових праць Харківського військового університету. – Х., 2006. – С. 225–233.
42. Гольдштейн А. Л. Исследование операций: многокритериальные задачи / А. Л. Гольдштейн. – М. : Наука, 1995.− 235 с.
43. Гоппа В. Д. Введение в алгебраическую теорию информации / В. Д. Гоппа. − М. : Наука. Физматлит, 1995. − 112 с.
44. Горбатов В. А. Фундаментальные основы дискретной математики / В. А. Горбатов // Информационная математика. – М. : Наука. Физматлит, 2000. − 544 с.
45. Гордон В. С. К вопросу минимизации функций на множестве перестановок частично упорядоченных элементов / В. С. Гордон, Я. М. Шафранский // Известия АН БССР. – 1979. – № 2. – С. 122–124. – (Серия физико-математических наук).
46. Гуляницкий Л. Ф. Метаэвристический метод деформируемого многогранника в комбинаторной оптимизации / Л. Ф. Гуляницкий, И. В. Сергиенко // Кибернетика и системный анализ. – 2007. − № 6. – С. 70–79.
47. Гуляницкий Л. Ф. Разработка гибридных методов дискретной оптимизации на основе G-алгоритмов / Л. Ф. Гуляницкий // Компьютерная математика. – 2005. − № 1. – С. 143–151.
48. Гуляницкий Л. Ф. Разработка гибридных методов дискретной оптимизации на основе G-алгоритмов / Л. Ф. Гуляницкий, Н. К. Тимофеева // Управляющие системы и машины. – 1982. − № 3. – С. 50–53.
49. Гуляницкий Л. Метаэвристический метод деформаций для решения задач комбинаторной оптимизации / Л. Гуляницкий // Knowledge. Dialogue. Solution (KDS – 2007) : XIII International Conference, (Varna, June 2007). – Sofia : ITHEA, 2007. – Vol. 1. − P. 95–102.

50. Гуляницкий Л. Ф. Об одном метаэвристическом методе комбинаторной оптимизации / Л. Ф. Гуляницкий // Компьютерная математика. – 2006. – № 2. – С. 1–6.
51. Гуляницкий Л. Ф. Метаэврический метод деформируемого многранника в комбинаторной оптимизации / Л. Ф. Гуляницкий, И. В. Сергиенко // Кибернетика и системный анализ. – 2007. – № 6. – С. 70–79.
52. Гуляницький Л. Ф. До формалізації та класифікації задач комбінаторної оптимізації / Л. Ф. Гуляницький // Теорія оптимальних рішень. – 2008. − № 7. – С. 45–49.
53. Гупал А. М. Оптимальные процедуры распознавания / А. М. Гупал, И. В. Сергиенко. – К. : Наукова думка, 2008. – 232 с.
54. Гупал А. М. Комплементарность оснований в хромосомах ДНК/ А. М. Гупал, А. А. Вагис // Проблемы управления и информатики. – 2005. – № 5. – С. 90–94.
55. Гэри М. Вычислительные машины и труднорешаемые задачи / пер. с англ. / М. Гэри, Д. Джонсон. – М. : Мир, 1982. – 416 с.
56. Донец Г. А. Алгоритм поиска значений линейной функции на лексикографически упорядоченных перестановках / Г. А. Донец, Л. Н. Колечкина // Теорія оптимальних рішень. – 2009 – № 8. – С. 3−8.
57. Донец Г. А. Локализация значения линейной функции заданной на перестановках / Г. А. Донец, Л. Н. Колечкина // Радиоэлектроника и информатика. – 2009. − № 1. – С. 76–81.
58. Донец Г. А. Метод упорядочения значений линейной функции на множестве перестановок / Г. А. Донец, Л. Н. Колечкина // Кибернетика и системный анализ – 2009. −№ 2. – С. 50−61.
59. Донец Г. А. Об одной задачи оптимизации дробно-линейной функции цели на перестановках / Г. А. Донец, Л. Н. Колечкина // Проблемы управления и информатики. – 2010. − № 2. – С. 12−16.
60. Донец Г. А. Об одном подходе к решению комбинаторной задачи оптимизации на графах / Г. А. Донец, Л. Н. Колечкина // Управляющие системы и машины. – 2009. − № 4. – С. 36−42.

61. Донец Г. А. Построение гамильтонова пути в графах перестановочных многогранников / Г. А. Донец, Л. Н. Колечкина // Кибернетика и системный анализ. – 2010. – № 1. – С. 10–16.
62. Донец Г. А. Экстремальные покрытия графов / Г. А. Донец, А. Я. Петренюк. – Кировоград : ОАО «Кіровоградське видавництво», 2009. – 170 с.
63. Донець Г. А. Метод моделювання структури вхідних даних і підкласи розв'язних задач / Г. А. Донець, Н. К. Тимофієва // Математичне та програмне забезпечення інтелектуальних систем (MPZIS – 2007). П'ята міжнар. наук-практ. конф. Дніпропетровськ, 14–16 листопада 2007 р. – Дніпропетровськ, 2007. – С. 52–53.
64. Дубов Ю. А. Многокритериальные модели формирования и выбора вариантов систем / Ю. А. Дубов, С. И. Травкин, В. Н. Якимец. – М. : Наука, 1986. – 296 с.
65. Емеличев В. А. Алгоритм линейной свертки в последовательной оптимизации критериев / В. А. Емеличев, О. А. Янушкевич. − Минск, 1996. − 28 с. Препринт / АН Беларуси. Ин-т техн. кибернетики; № 5).
66. Емеличев В. А. Лексикографические оптимумы многокритериальной задачи дискретной оптимизации / В.А. Емеличев, М. К. Кравцов, О. А. Янушкевич // Мат. заметки. − 1995. − Т. 58. − вып. 3. − С. 365–371.
67. Емеличев В. А. Лекции по теории графов / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. − М. : Наука, 1990. – 384 с. (Изд. 2, испр. – М. : УРСС, 2009. – 392 с.).
68. Емеличев В. А. Линейная свертка критериев в задачах лексикографической оптимизации / В. А. Емеличев, О. А. Янушкевич // Докл. АН Беларуси. − 1999. − Т. 43. − № 2. − С. 29–32.
69. Емеличев В. А. Многокритериальные задачи об остовах графа / В. А. Емеличев, В. А. Перепелица // Докл. АН СССР. − 1988. − Т. 298. − № 3. − С. 544–547.
70. Емеличев В. А. О задачах векторной дискретной оптимизации на системах подмножеств, неразрешимых с помощью алгоритмов линейной свертки критериев / В. А. Емеличев, М. К. Кравцов // Журн. вычисл. математики и мат. физики. − 1994. − Т. 34. − № 7. − С. 1082–1094.

71. Емеличев В. А. О задачах лексикографической оптимизации / В. А. Емеличев, О. А. Янушкевич // Дискрет. анализ и исслед. операций. Сер. 1. − 1998. − Т. 5. − № 4. − С. 30–37.
72. Емеличев В. А. О многокритериальных задачах нахождения лексикографических оптимумов / В. А. Емеличев, А. А. Гладкий, О. А. Янушкевич // Изв. АН Беларуси. Сер. физ.-мат. наук. − 1996. − № 3. − С. 82–86.
73. Емеличев В. А. О разрешимости одного класса дискретных векторных задач с помощью алгоритма линейной свертки критериев / В. А. Емеличев, М. К. Кравцов, О. А. Янушкевич // Журн. вычисл. математики и мат. физики. − 1997. − Т. 37. − № 11. − С. 1404–1408.
74. Емеличев В. А. Об одном типе устойчивости многокритериальной задачи целочисленного линейного программирования в случае монотонной нормы / В. А. Емеличев, К. Г. Кузьмин // Изв. РАН. Теория и системы управления. – 2007. − № 5. – С. 45–51.
75. Емеличев В. А. Общий подход к исследованию устойчивости парето-оптимального решения векторной задачи целочисленного линейного программирования / В. А. Емеличев, К. Г. Кузьмин // Дискретная математика. – 2007. – Вып. 3. − С. 79–83.
76. Емеличев В. А. Разрешимость векторной траекторной задачи на «узкие места» с помощью алгоритма линейной свертки критериев / В. А. Емеличев, М. К. Кравцов, О. А. Янушкевич // Докл. АН Беларуси. − 1996. − Т. 40. − № 4. − С. 29–33.
77. Емеличев В. А. Сложность дискретных многокритериальных задач / В. А. Емеличев, В. А. Перепелица // Дискрет. математика. – 1994. – Вып. 1. – С. 3–33.
78. Емеличев В. А. Условия Парето-оптимальности в дискретных векторных задачах оптимизации / В. А. Емеличев, О. А. Янушкевич // Дискретная математика. − 1997. − Т. 9. – Вып. 3. − С. 153–160.
79. Емеличев В. А. О неразрешимости векторных задач дискретной оптимизации на системах подмножеств в классе алгоритмов линейной свертки критериев / В.А. Емеличев, М.К. Кравцов // Докл. РАН. − 1994. − Т. 334. − № 1. − С. 9–11.

80. Емеличев В. А. О некоторых алгоритмических проблемах многокритериальной оптимизации на графах / В. А. Емеличев, В. А. Перепелица // Журн. вычисл. математики и мат. физики. – 1989. – Т. 29. – № 2. – С. 171–183.
81. Емеличев В. А. О радиусе устойчивости векторной задачи целочисленного линейного программирования в случае регулярности нормы в критериальном пространстве / В. А. Емеличев, К. Г. Кузьмин // Кибернетика и системный анализ. – 2010. − № 1. – С. 82–90.
82. Емеличев В. А. К вычислительной сложности дискретных многокритериальных задач / В. А. Емеличев, В. А. Перепелица // Изв. АН СССР. Техн. кибернетика. − 1988. − № 1. − С. 78–85.
83. Емеличев В. А. Лексикографические оптимумы многокритериальной задачи / В. А. Емеличев, Э. Гирлих, О. А. Янушкевич // Дискрет. анализ и исслед. операций Сер. 1. − 1997. − Т. 4. − № 2. − С. 3–14.
84. Емеличев В. А. О некоторых алгоритмических проблемах многокритериальной оптимизации на графах / В. А. Емеличев, В. А. Перепелица // Журн. вычисл. математики и мат. физики. − 1989. − Т. 29. − № 2. − С. 171–183.
85. Емеличев В. А. Условия эффективности решения векторной задачи оптимизации / В. А. Емеличев, А. В. Пашкевич, О. А. Янушкевич // Дискретная математика. − 1999. − Т. II. − Вып. 1. − С. 140–145.
86. Емеличев В. А. Дискретная оптимизация. Последовательные схемы решения I, II / В. А. Емеличев // Кибернетика. – 1971. – № 6. – С. 109–121 ; 1972. – № 2. – С. 92–103.
87. Емеличев В. А. Многогранники, графы, оптимизация / В. А. Емеличев, М. М. Ковалев, М. К. Кравцов. – М. : Наука, 1981. – 342 с.
88. Емеличев В. А. Сложность дискретных многокритериальных задач / В. А. Емеличев, В. А. Перепелица // Дискретная математика. – 1994. – Вып. 1. – С. 3–33 ; 1994.− Вып. 6. – С. 5−32.
89. Емец О. А. Комбинаторная оптимизация на размещениях / О. А. Емец, Т Н. Барболина. – К. : Наукова думка, 2008. – 159 с.
90. Емец О. А. Об экстремальных свойствах недифференцируемых выпуклых функций на евклидовом множестве

сочетаний с повторениями / О. А. Емец // Український математичний журнал. – 1994. – Т 46. – № 6. – С. 680–691.
91. Емец О. А. Решение задач оптимизации с дробно-линейными целевыми функциями и дополнительными линейными ограничениями на перестановках / О. А. Емец, Л. Н. Колечкина // Кибернетика и системный анализ. – 2004. − № 3. – С. 156−169.
92. Еремин И. И. Введение в теорию линейного и выпуклого программирования / И. И. Еремин, Н. И. Астафьев. – М. : Наука, 1976. – 192 с.
93. Еремин И. И. Несобственные задачи линейного и выпуклого программирования / И. И. Еремин, В. Д. Мазуров, Н. И. Астафьев. – М. : Наука, 1983. –336 с.
94. Ермольев Ю. М. Математические методы исследования операций / Ю. М. Ермольев, И. И. Ляшко, В. С. Михалевич. – К. : Вища школа, 1979. – 312 с.
95. Ермольев Ю. М. Экстремальные задачи на графах / Ю. М. Ермольев, И. М. Мельник. – К. : Наукова думка, 1970. – 175с.
96. Ємець О. О. Деякі операції та відношення над нечіткими числами / О. О. Ємець, Ол-ра О.Ємець // Наукові вісті НТУУ «КПІ». – 2008. – № 5. – С. 39–46.
97. Ємець О. О. Задача оптимізації на переставленнях з дробово-лінійною цільовою функцією: властивості множини допустимих розв'язків / О. О. Ємець, Л. М. Колєчкіна // Український математичний журнал. – 2000. – Т. 52. – № 12. – С. 1630–1640.
98. Ємець О. О. Задачі комбінаторної оптимізації з дробово-лінійними цільовими функціями / О. О. Ємець, Л. М. Колєчкіна. – К. : Наукова думка, 2005. – 118 с.
99. Ємець О. О. Задачі оптимізації на полікомбінаторних множинах : властивості та розв'язування / О. О. Ємець, О. В. Роскладка. – Полтава : РВВ ПУСКУ, 2006. – 130 с.
100. Ємець О. О. Розв'язування багатокритеріальної задачі з лінійними критеріями на множині переставлень / О. О. Ємець, Л. М. Колєчкіна, А. М. Нагірна // Вісник Хмельницького національного університету. – 2005. − № 5. − С. 345−347.

101. Жуковин В. Е. Нечеткие многокритериальные модели принятия решений / В. Е. Жуковин. – Тбилиси, 1988. – 231 с.
102. Жуковский В. И. Многокритериальное принятие решений в условиях неопределенности / В. И. Жуковский, В. С. Молоствов. – М. : Изд-во МНИИПУ, 1988. – 267 с.
103. Жуковский В. И. Риск многокритериальных и конфликтных систем при неопределенности / В. И. Жуковский, Л. В. Жуковская. – М. : Эдиториал УРСЭ, 2004. – 272 с.
104. Журавлев Ю. И. Избранные научные труды / И. Ю. Журавлев. – М. : Магистр, 1998. – 420 с.
105. Зайченко Ю. П. Исследование операций. Нечеткая оптимизация : учеб. пособие / Ю. П. Зайченко. – К. : Вища школа, 1991. – 198 с.
106. Зыков А. А. Основы теории графов / А. А. Зыков. – М. : Вузовская книга, 2004. – С. 664.
107. Золотых Н. Ю. Параллельный алгоритм нахождения общего решения системы линейных неравенств / Н. Ю. Золотых, С. С. Лялин // Вестник ННГУ. – 2009. – № 5. – С. 193–199.
108. Кини Р. Л. Принятие решений при многих критериях: предпочтения и замещения / Р. Л. Кини, Х. Райфа. – М. : Радио и связь, 1981. – 560 с.
109. Кирсанов М. Н. Графы в Maple / М. Н. Кирсанов. − М. : Физматлит, 2007. − 168 с.
110. Кондрук Н. Е. Алгоритм кластеризації критеріального простору для задач вибору / Н. Е. Кондрук, М. М. Маляр // Вісник Київського університету. Серія : фіз.-мат. наук, Вип. 3. – К., 2006.
111. Ковалев М. М. Матроиды в дискретной оптимизации / М. М. Ковалев. – Минск : Университетское, 1987. –220 с.
112. Ковалев М. М. Дискретная оптимизация / М. М. Ковалев. – Минск : Изд-во БГУ, 1977. – 192 с.
113. Когут П.І. Про розв’язність одного класу задач векторної оптимізації / П.І. Когут, І.В. Нечай // Наукові вісті. – 2006. − № 5. – С. 148-158.
114. Козерацкая Л. Н. Задачи дискретной оптимизации: исследование устойчивости / Л. Н. Козерацкая, Т. Т. Лебедева, И. В. Сергиенко // Обозрение прикладной и промышленной математики. – 1995. – № 1. – С. 12–30.

115. Козин И. В. Принципы симметрии в теории принятия решений / И. В. Козин. – Запорожье : Полиграф, 2008. – 164 с.
116. Колечкина Л. Н. Многокритериальные задачи на комбинаторном множестве полиразмещений : структурные свойства решений / Л. Н. Колечкина // Artificial Intelligence and Decision Making : Supplement to International Journal «Information Technologies and Knowledge». – 2008. – Vol. 2. − P. 180–186. – (Intern. Book Series «Information science and computing» ; № 7).
117. Колечкина Л. Н. Многокритериальные комбинаторные задачи оптимизации на множестве полиразмещений / Л. Н. Колечкина, Е. А. Родионова // Кибернетика и системный анализ. – 2008. – № 2. – С. 152–160.
118. Колечкина Л. Н. Моделирование прикладных задач векторными задачами на комбинаторных конфигурациях / Л. Н. Колечкина, Е. А. Родионова // Радиоэлектроника и информатика. – 2009. – № 3. − С. 62−68.
119. Колечкина Л. Н. О нахождении Парето-оптимальных решений в многокритериальных комбинаторных задачах на множестве размещений / Л. Н. Колечкина // Теорія оптимальних рішень. – 2008. – № 7. – С. 109–116.
120. Колечкина Л. Н. Об одном алгоритме решения комбинаторных задач / Л. Н. Колечкина // Комп’ютерна математика в науці, інженерії та освіті : матеріали третьої міжнар. наук.-технічної конф., (Полтава, 1−31 жовт. 2009 р.). – К. : Вид-во НАНУ, 2009. − С. 19.
121. Колечкина Л. Н. Об одном алгоритме решения комбинаторных задач векторной оптимизации на множестве размещений / Л. Н. Колечкина // Искусственный интелект. – 2010 – № 1. – С. 61–69.
122. Колечкина Л. Н. Обоснование структурированного метода локализации значения линейной функции, заданной на комбинаторной конфигурации перестановок / Л. Н. Колечкина // Динамические системы. – 2009. − № 27. – С. 1−13.
123. Колечкина Л. Н. Оптимальные решения многокритериальных комбинаторных задач на размещениях / Л. Н. Колечкина // Теорія оптимальних рішень. – 2007. – № 6. – С. 67–73.

124. Колєчкіна Л. М. Огляд комбінаторних задач та підходів до їх розв'язання за допомогою графів / Л. М. Колєчкіна, С. Є. Швачко // Економічне відродження України : матеріали VI міжнар. наук.-практ. конф., 22 трав. 2009 р. − К. : Міжнародний науково-технічний університет ім. академіка Юрія Бугая, 2009. – С. 238−239.
125. Колєчкіна Л. М. Властивості задач багатокритеріальної оптимізації на комбінаторних множинах та методи їх розв'язання / М. Л. Колєчкіна. – Полтава : РВВ ПУСКУ, 2008. – 162 с.
126. Колєчкіна Л. М. Моделювання прикладних задач багатокритеріальними комбінаторними задачами на поліпереставках / Л. М. Колєчкіна, О. А. Родіонова // Волинський математичний вісник. – 2009. − Вип. 6. − С. 72−86. – (Серія «Прикладна математика»).
127. Колєчкіна Л. М. Моделювання та розв'язування економічних задач оптимізації відносних показників з урахуванням комбінаторних властивостей розв'язку / Л. М. Колєчкіна, А. М. Нагірна // Наукові вісті Нацаціонального технічного університету України «Київський політехнічний інститут». – 2006. – № 5. – С. 34–40.
128. Колєчкіна Л. М. Питання розв'язування комбінаторних багатокритеріальних задач на множині полі розміщень / Л. М. Колєчкіна, О. А. Родіонова // Наука – практика – освіта : матеріали V міжвуз.наук.-практ. конф., 23 трав. 2008 р. / упоряд. Л. Т. Коломієць. – К. : ЗАТ «ДОРАДО», 2008. – С. 153−157.
129. Колєчкіна Л. М. Постановка задачі багатокритеріальної комбінаторної оптимізації на полірозміщеннях та підхід до розв'язання / Л. М. Колєчкіна, О. А. Родіонова // Радиоэлектроника и информатика. – 2007. – №. 1. – С. 84–88.
130. Колчин В. Ф. Случайные графы / В. Ф. Колчин. – М. : ФИЗМАТЛИТ, 2004. – 208 с.
131. Кондрук Н. Е. Алгоритм кластеризації критеріального простору для задач вибору / Н. Е. Кондрук, М. М. Маляр // Вісник Київського університету. – 2006. − Вип. 3.− С. 124–128. − (Серія фізико – мататематичних наук).
132. Корбут А. А. Дискретное программирование / А. А. Корбут, Ю. Ю. Финкельштейн. – М. : Наука, 1969. – 368 с.

133. Корбут А. А. Метод ветвей и границ: Обзор теории, алгоритмов, программ и приложений / А. А. Корбут, И. Х. Сигал, Ю. Ю. Филькенштейн // Math. Operationsforsch und Statist. − 1977. – Т. 8, № 2. – С. 253–280. – (Ser. Optimiz.).
134. Кормен Т. Х. и др. Часть VI. Алгоритмы для работы с графами / Т. Х. Кормен и др. //Алгоритмы: построение и анализ = Introduction to Algorithms. Т. Кормен − 2-е изд. − М. : Вильямс, 2006. − С. 1296.
135. Кофман А. Введение в теорию нечетких множеств / А. Кофман. – М. : Радио и связь, 1982. – 432 с.
136. Кравцов М. К. Линейная свертка критериев в бикритериальной оптимизации / М. К. Кравцов, О. А. Янушкевич // Изв. вузов. Математика. − 1998. − № 12. − С. 63–70.
137. Кравцов М. К. Неразрешимость векторной дискретной оптимизации в классе алгоритмов линейной свертки критериев / М. К. Кравцов // Дискретная математика. − 1996. – Т. 8. – Вып. 2. − С. 89–96.
138. Кравцов М. К. О разрешимости векторной задачи с помощью алгоритма линейной свертки критериев / М. К. Кравцов, О. А. Янушкевич // Математические заметки. − 1997. − Т. 62. – Вып. 4. − С. 502–509.
139. Крейнес М. Г. Алгоритм решения некоторого класса дискретных многокритериальных задач / М. Г. Крейнес, Н. М. Новиков // Журнал вычислительной математики и математической физики. – 1983. – Т. 23. – № 3. − С. 214–218.
140. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. − М. : Мир, 1978. – 429 с.
141. Кук Д. Компьютерная математика / пер. с англ. Д. Кук, Г. Бейз. – М. : Наука, Гл. ред. физ.-мат. лит., 1990. – 384 с.
142. Ларичев О. И. Наука и искусство принятия решений / О. И. Ларичев. – М. : Наука, 1979. – 200 с.
143. Лебедев Б. Д. Задача оптимизации по упорядоченной совокупности критериев / Б. Д. Лебедев, В. В. Подиновский, Р. С. Стырикович // Экономика и математические методы. − 1971. − Т. 7. – Вып. 4. − С. 612–616.
144. Лебедев Б. Д. Задача оптимизации по упорядоченной совокупности критериев / Б. Д. Лебедев, В. В. Подиновский, Р. С. Стырикович // Экономика и математичесие методы. − 1971. − Т. 7. – Вып. 4. − С. 612–616.

145. Лебедева Т. Т. Устойчивость векторных задач целочисленной оптимизации: взаимосвязь с устойчивостью множеств оптимальных и неоптимальных решений / Т. Т. Лебедева, Н. В. Семенова. Т. И. Сергиенко // Кибернетика и системный анализ. – 2005. – № 4 – С. 90–100.
146. Лебєдєва Т. Т. Умови оптимальності та розв'язуваності в задачах лінійної векторної оптимізації з опуклою допустимою множиною / Т. Т. Лебєдєва, Н. В. Семенова, Т. І. Сергієнко // Доповіді НАНУ. – 2003. – № 10. – С. 80–85.
147. Левитская А. А. Одна комбинаторная задача в классе перестановок над кольцом Zn вычетов по нечетному модулю n / А. А. Левитская // Проблемы управления и информатики. – 1996. – № 5. – С. 99–108.
148. Липский В. Комбинаторика для программистов / В. Липский. – М. : Мир, 1988. – 200 с.
149. Лихтенштейн В. Е. Модели и методы дискретного программирования / В. Е. Лихтенштейн. – М. : Наука, 1997. – 240 с.
150. Лэсдон Л.С. Оптимизация больших систем / Л. С. Лэсдон. – М. : Наука, 1975. – 432 с.
151. Ляшенко И. Н. Взаимодействия экономики и окружающей среды (эколого-экономическое моделирование) / монография «Экономика и кибернетика в начале XXI века» под научной редакцией Задорожнего Г. В., Михайленко В. Г. – Х. : ХНУ, 2005 – 260 с.
152. Ляшенко І. М. Економіко-математичні методи та моделі сталого розвитку / І. М. Ляшенко. – К. : Вища школа, 1999. – 236 с.
153. Ляшенко І. М. Моделювання економічних, екологічних та соціальних процесів / І. М. Ляшенко, М. В. Коробов, І. А. Горіцина. – К. : ВПЦ «Київський університет», 2009. – 320 с.
154. Ляшенко І. М. Основи математичного моделювання економічних, екологічних та соціальних процесів / І. М. Ляшенко, М. В. Коробов, А. М. Столяр. – Т. : Навчальна книга – Богдан, 2006. – 304 с.
155. Максишко Н. К. Моделі та методи розв'язання прикладних задач покриття на графах та гіперграфах / Н. К. Максишко, Т. В. Заховалко. – Запорожье : Полиграф, 2009. – 244 с.

156. Машунин Ю. К. Методы и модели векторной оптимизации / Ю. К. Машунин. – М. : Наука, 1986. – 140 с.
157. Меламед И. И. Исследование линейной свертки критериев в многокритериальном дискретном программировании / И. И. Меламед, И. Х. Сигал // Журн. вычисл. математики и мат. физики. − 1995. − Т. 35. − № 8. − С. 1260–1270.
158. Меламед И. И. Вычислительное исследование трехкритериальных задач о деревьях и назначениях / И. И. Меламед, И. Х. Сигал // Журн. вычисл. математики и мат. физики. – 1998. – Т. 38. – № 10. – С. 1780–1787.
159. Меламед И. И. Линейная свертка критериев в многокритериальной оптимизации / И. И. Меламед // Автоматика и телемеханика. − 1997. − № 9. − С. 119–125.
160. Мину М. Математическое программирование. Теория и алгоритмы / М. Мину.– М. : Наука, 1990. – 488 с.
161. Миркин Б. Г. Проблемы группового выбора / Б. Г. Миркин. – М. : Наука, 1974. – 256 с.
162. Михалевич В. С. Вычислительные методы исследования и проектирования сложных систем / В. С. Михалевич, В. Л. Волкович. – М. : Наука, 1982. – 286 с.
163. Михалевич В. С. Оптимизационные задачи производственно-транспортного планирования: модели, методы, алгоритмы / В. С. Михалевич, В. А. Трубин, Н. З. Шор. – М. : Наука, 1986. – 264 с.
164. Михалевич В. С. Последовательные схемы оптимизации в задачах упорядочения выполнения работ / В. С. Михалевич, В. В. Шкурба // Кибернетика. – 1966. – № 2. – С. 34–40.
165. Михалевич М. В. Моделирование переходной экономики. Модели, методы, информационные технологии / М. В. Михалевич, И. В. Сергиенко. – К. : Наукова думка, 2005. – 671 с.
166. Модели и методы оптимизации надежности сложных систем / Волкович В. Л. В и др. – К. : Наукова думка, 1993. – 312 с.
167. Мулен Э. Кооперативное принятие решений: аксиомы и модели : [пер. с англ.] / Э. Мулен. – М. : Мир, 1991. – 464 с.
168. Недашковский Н. А. Вычислительные алгоритмы для линейных балансовых моделей межотраслевого эколого-

экономического взаимодействия / Н. А. Недашковский, Т. И. Крошка // Кибернетика и системный анализ. – 2010. − № 1. – С. 17–29.
169. Нечеткие множества в моделях управления и искусственного интеллекта / А. Н. Аверкин, И. З. Батыршин, А. Ф. Блишун и др. ; под ред. Д. А. Поспелова. – М. : Наука, 1986. – 312 с.
170. Нечеткие множества и теория возможностей. Последние достижения / под ред. Р. Р. Ягера. – М. : Радио и связь, 1986. – 408 с.
171. Ногин В. Д. Логическое обоснование принципа Эджворта-Парето / В. Д. Ногин // Журнал вычислительной математики и математической физики. – 2002. – Т. 42. – № 7.− С. 951–957.
172. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход / В. Д. Ногин. – М. : Физматлит, 2002. – 144 с.
173. Нурлыбаев А. Н. О графе перестановочного многогранника / А. Н. Нурлыбаев // Доклады АН Республики Казахстана. – 1992. – № 3. – С. 14–20.
174. Нурминский Е. А. Проекция на полиэдр во внешнем представлении / Е. А. Нурминский // ЖВМ и МФ, 2008.
175. Обработка нечеткой информации в системах принятия решений / А. Н. Борисов, А. В. Алексеев, Г. В. Меркурьева и др. – М. : Радио и связь, 1989. – 304 с.
176. Озерной В. М. Методология решения дискретных многокритериальных задач / В. М. Озерной, М. Г. Гафт // Многокритериальные задачи принятия решений. – М. : Машиностроение, 1978. – С. 14–47.
177. Оре О. Теория графов. / О. Оре. − 2-е изд. − М. : Наука, 1980. − С. 336.
178. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации / С. А. Орловский. – М. : Наука, 1981. – 208 с.
179. Павлов А. А. Особенности решения NP-трудных задач комбинаторной оптимизации / А. А. Павлов, Инхуэйн Ван // Інформатика та нові технології. – 1997. − № 1. – С. 13.
180. Павлов О. А. Принцип розпаралелювання обчислень як засіб підвищення ефективності ПДС-алгоритмів для важкорозв’язувальних комбінаторних задач / О. А. Павлов,

Л. О. Павлова // Наукові вісті НТУУ «КПІ». – 1997. – № 1. – С. 22–26.
181. Палагин А. В. Байесовские процедуры распознавание литературных текстов / А. М. Гупали др. // Проблемы управления и информатики. – 2009. – № 2. – С. 82–89.
182. Панишев А. В. Методы и модели оптимизации в проблеме коммивояжера / А. В. Панишев, Д. Д. Плечистый. – Житомир : ЖДТУ, 2006. – 306 с.
183. Панішев А. В. Вступ до теорії складності дискретних задач / А. В. Панішев, О. М. Данильченко, В. О. Скачков. – Житомир : Житомир. держ. технолог. ун-т., 2004. – 236 с.
184. Пападимитриу Х. Комбинаторная оптимизация. Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985. – 512 с.
185. Парасюк И. Н. О развитии метода вектора спада на случай размытых окрестностей / И. Н. Парасюк, М. Ф. Каспшицкая // Компьютерная математика. – 2008. – № 2. – С. 145−155.
186. Парасюк И. Н. О решении комбинаторной многокритериальной оптимизационной задачи нечетким методом вектора спада / И. Н. Парасюк, М. Ф. Каспшицкая // Компьютерная математика. – 2009. – № 2. – С. 150−158.
187. Перепелица В. А. Исследование одного класса целочисленных многокритериальных задач / В. А. Перепелица, И. В. Сергиенко // Журнал вычислительной математики и математической физики. – 1988. – Т. 28. – № 3. – С. 400–419.
188. Перепелица В. А. К проблеме нахождения множеств альтернатив в дискретных многокритериальных задачах / В. А. Перепелица, И. В. Сергиенко // Кибернетика. – 1987. – № 5. – С. 85–93.
189. Перепелица В. А. Полиномиальные и NP-полные многокритериальные задачи перечисления альтернатив / В. А. Перепелица, И. В. Сергиенко // Теория и программная реализация методов дискретной оптимизации. – К. : Ин-т кибернетики им. В. М. Глушкова, 1989. – С. 58–69.
190. Побудова і дослідження методів та інформаційних технологій розв’язання оптимізаційних дискретних задач великої розмірності / І. В. Сергієнко та ін. // Фундаментальні

орієнтири науки. Математика, інформатика, механіка : зб. статей за матеріалами проектів ДФФД. – К. : Академперіодика, 2005. – С. 23–42.
191. Подиновский В. В. Оптимизация по последовательно применяемым критериям / В. В. Подиновский, В. М. Гаврилов. – М. : Сов. радио, 1975. – 192 с.
192. Подиновский В. В. Парето-оптимальные решения многокритериальных задач / В. В. Подиновский, В. Д. Ногин. – М. : Наука, 1982. – 256 с.
193. Полищук Л. И. Анализ многокритериальных экономико-математических моделей / Л. И. Полищук. – М. : Наука, 1989. – 182 с.
194. Поспелов Г. С. Процедуры и алгоритмы формирования комплексных программ / Г. С. Поспелов, В. А. Ириков, А. Е. Курилов. – М. : Наука, 1985. – 424 с.
195. Пшеничный Б. Н. Комбинаторный метод решения общей задачи выпуклого программирования / Б. Н. Пшеничный, Э. И. Ненахов, В. Н. Кузьменко // Кибернетика и системный анализ. – 1998. – № 4 – С. 121–134.
196. Пшеничный Б. Н. Численные методы в экстремальных задачах / Б. Н. Пшеничный, Ю. М. Данилин. – М. : Наука, 1975. – 287 с.
197. Рейнгольд Э. Комбинаторные алгоритмы: теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. − М. : Мир, 1980. − 476 с.
198. Реклейтис Г. Оптимизация в технике : в 2 т. / Г. Реклейтис, А. Рейвиндран, К. Рэгедел. – М. : Мир, 1986.
Т 1. – 1986. − 352 с.
Т 2. – 1986. − 364 с.
199. Риордан Дж. Введение в комбинаторный анализ / Дж. Риордан. − М. : Иностранная литература, 1963. – 288 с.
200. Рокафеллар Р. Выпуклый анализ / Р. Рокафеллар. – М. : Мир, 1973. – 470 с.
201. Романовский И. В. Дискретный анализ / И. В. Романовский. – С.Пб. : Невский диалект, 2000. – 240 с.
202. Рощин В. А. Вопросы решения и исследования одного класса задач неточного целочисленного программирования / В. А. Рощин, Н. В. Семенова, И. В. Сергиенко // Кибернетика. – 1989. – № 2. – С. 42–47.

203. Рыбников К. А. Введение в комбинаторный анализ / К. А. Рыбников. – М. : Изд-во Моск. ун-та, 1985. – 308 с.
204. Рыбников К. А. Комбинаторный анализ. Задачи и упражнения / К. А. Рыбников. – М. : Изд-во Моск. ун-та, 1982. – 368 с.
205. Салий В. Н. Алгебраические основы теории дискретных систем / В. Н. Салий, А. М. Богомолов. − М. : Физико-математическая литература, 1997.
206. Сачков В. Н. Комбинаторные методы дискретной математики / В. Н. Сачков. – М. : Наука, 1977. – 320 с.
207. Сачков В. Н. Введение в комбинаторные методы дискретной математики / В. Н. Сачков. – М. : Наука. Главная редакция физико-математической литературы, 1982. – 384 с.
208. Свами М. Графы, сети и алгоритмы / М. Свами, К. Тхулалирман. – М. : Мир, 1984. – 455 с.
209. Семенова Н. В. Векторні задачі дискретної оптимізації на комбінаторних множинах: методи їх дослідження та розв’язання / Н. В. Семенова, Л. М. Колєчкіна. – К. : Наукова думка, 2009. – 262 с.
210. Семенова Н. В. Векторные задачи на комбинаторном множестве полиразмещений: условия оптимальности и подход к решению / Н. В. Семенова // Artificial Intelligence and Decision Making : Supplement to International Journal «Information Technologies and Knowledge». – 2008. – Vol. 2. − P. 187–195.
211. Семенова Н. В. Гарантирующие и оптимистические решения задач целочисленной оптимизации с выпуклыми квадратичными функциями ограничений / Н. В. Семенова // Теорія оптимальних рішень. – 2006. – № 5.– С. 39– 46.
212. Семенова Н. В. Методы поиска гарантирующих и оптимистических решений задач целочисленной оптимизации в условиях неопределенности данных / Н. В. Семенова // Кибернетика и системный анализ. – 2007. –№ 1. – С. 103–114.
213. Семенова Н. В. Многокритериальные задачи на комбинаторном множестве полиразмещений: полиэдральный подход к их решению / Н. В. Семенова, Л. Н. Колечкина // Кибернетика и системный анализ. – 2009. – № 3. – С. 118–126.

214. Семенова Н. В. О решении векторных задач частично дискретной оптимизации / Н. В. Семенова // Комп'ютерна математика. – 2008. – № 2.– С. 156–164.
215. Семенова Н. В. Подход к решению векторных задач дискретной оптимизации на комбинаторном множестве перестановок / Н. В. Семенова, Л. Н. Колечкина, А. Н. Нагорная // Кибернетика и системный анализ – 2008. – № 3 – С. 158–172.
216. Семенова Н. В. Поліедральний підхід до розв'язання одного класу векторних задач комбінаторної оптимізації / Н. В. Семенова, Л. М. Колєчкіна, А. М. Нагірна // Доповіді НАН України. – 2009. – № 6. – С. 46–53.
217. Семенова Н. В. Решение и исследование векторных задач комбинаторной оптимизации на множестве полиперестановок / Н. В. Семенова, Л. Н. Колечкина, А. Н. Нагорная // Проблемы управления и информатики. – 2008. – № 6. – С. 26–41.
218. Семенова Н. В. Розв'язання багатокритеріальних задач оптимізації на множині поліперестановок / Н. В. Семенова, Л. М. Колєчкіна, А. М. Нагірна // Доповіді НАН України. – 2009. – № 2. – С. 41– 48.
219. Семенова Н. В. Розв'язування задач векторної оптимізації з дробово-лінійними функціями критеріїв на комбінаторній множині полірозміщень / Н. В. Семенова, Л. Н. Колечкина, А. Н. Нагорная // Наукові вісті Національного технічного університету України «Київський політехнічний інститут». – 2009. – № 2. – С. 53–60.
220. Семенова Н. В. Условия оптимальности в векторных задачах комбинаторной оптимизации / Н. В. Семенова // Теорія оптимальних рішень. − 2008. – № 7. – С. 153–160.
221. Семенова Н. В.Об одном подходе к решению векторных задач с дробно-линейными функциями критериев на комбинаторном множестве размещений / Н. В. Семенова, Л. Н. Колечкина, А. Н .Нагорная // Проблемы управления и информатики. – 2010. – № 1. – С. 131–144.
222. Сергиенко И. В. Исследование устойчивости и параметрический анализ дискретных оптимизационных задач / И. В. Сергиенко, Л. Н. Козерацкая, Т. Т. Лебедева. – К. : Наукова думка, 1995. − 170 с.

223. Сергиенко И. В. Методы предсказания пространственной структуры белков / И. В. Сергиенко и др. // Кибернетика и системный анализ. – 2010. − № 1. – С. 38–59.
224. Сергиенко И. В. О проблеме отыскания множества альтернатив в дискретных многокритериальных задачах / И. В. Сергиенко, В. А. Перепелица // Кибернетика. – 1987. − № 5. – С. 85–93.
225. Сергиенко И. В. Предсказание вторичной структуры белков на основе байесовских процедур распознавания на цепях Маркова / И. В. Сергиенко и др. // Кибернетика и системный анализ. – 2007. − № 2. – С. 59–64.
226. Сергиенко И. В. Задачи дискретной оптимизации: проблемы, методы решения, исследования / И. В. Сергиенко, В. П. Шило. – К. : Наукова думка, 2003. – 264 с.
227. Сергиенко И. В. Задачи целочисленного программирования с неоднозначно заданными данными: точные и приближенные решения / И. В. Сергиенко, Н. В. Семенова // Кибернетика и системный анализ. – 1995. – № 6. – С. 75–86.
228. Сергиенко И. В. Исследование устойчивости и параметрический анализ дискретных оптимизационных задач / И. В. Сергиенко, Л. Н. Козерацкая, Т. Т. Лебедева. – К. : Наукова думка, 1995. – 171 с.
229. Сергиенко И. В. Математические модели и методы решения задач дискретной оптимизации / И. В. Сергиенко. – К. : Наукова. думка, 1988. – 471 с.
230. Сергиенко И. В. Метод неявного перебора для решения некоторых экстремальных задач на множествах / И. В. Сергиенко, В. А. Рощин // Обчислювальна та прикладна математика. – 1996. – Вип. 80. – С. 106–118. – (Серія «Оптимізація»).
231. Сергиенко И. В. Модели и методы решения на ЭВМ комбинаторных задач оптимизации / И. В. Сергиенко, М. Ф. Каспшицкая. – К. : Наукова думка, 1981. – 288 с.
232. Сергиенко И. В. Некоторые задачи целочисленного программирования с неоднозначно заданными данными и их решение / И. В. Сергиенко, В. А. Рощин, Н. В. Семенова // Проблемы управления и информатики. – 1998. – № 6. – С. 116–123.

233. Сергиенко И. В. О существовании решений в задачах векторной оптимизации / И. В. Сергиенко, Т. Т. Лебедева, Н. В. Семенова // Кибернетика и системный анализ. – 2000. – № 6. – С. 39–46.
234. Сергиенко И. В. Системный анализ многокомпонентных распределенных систем / И. В. Сергиенко, В. С. Дейнека. – К. : Наукова думка, 2009. – 640 с.
235. Сергієнко І. В. Інформатика та комп’ютерні технології / І. В. Сергієнко. – К. : Наукова думка, 2004. – 432 с.
236. Сергієнко І. В. Інформатика в Україні: становлення, розвиток, проблеми / І. В. Сергієнко. – К. : Наукова думка, 1999. – 354 с.
237. Сигал И. Х. Параметризация приближенных алгоритмов решения некоторых классов задач дискретной оптимизации большой размерности / И. Х. Сигал // Известия РАН. Теория и системы управления. – 2002. – № 6. – С. 63–72.
238. Сигал И. Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы / И. Х. Сигал, А. П. Иванова. – М. : Физматлит, 2003. – 240 с.
239. Соболь И. М. Выбор оптимальных параметров в задачах со многими критериями / И. М. Соболь, Р. Б. Статников. – М. : Наука, 1981. – 107с.
240. Современное состояние теории исследования операций / под ред. Н. Н. Моисеева. – М. : Наука, 1979. – 464 с.
241. Стенли Р. Перечислительная комбинаторика / Р. Стенли. – М. : Мир, 1990. – 440 с.
242. Стоян Ю. Г. Математические модели и оптимизационные методы геометрического проектирования / Ю. Г. Стоян, С. В. Яковлев. – К. : Наукова думка, 1986. – 268 с.
243. Стоян Ю. Г. Оптимізація на поліроміщеннях: теорія та методи / Ю. Г. Стоян, О. О. Ємець, Є. М. Ємець. – Полтава : РВВ ПУСКУ, 2005. – 104 с.
244. Стоян Ю. Г. Решение некоторых многоэкстремальных задач методом сужающихся окрестностей / Ю. Г. Стоян, В. З. Соколовский. – К. : Наукова думка, 1980. – 205 с.
245. Стоян Ю. Г. Теорія і методи евклідової комбінаторної оптимізації / Ю. Г. Стоян, О. О. Ємець. – К. : Ін-т системних досліджень освіти, 1993. – 188 с.

246. Схрейвер А. Теори линейного и целочисленного программирования : в 2 т. / А. Схвейер. – М. : Мир, 1991.
Т. 1. – 1991. – 362 с.
Т. 2. – 1991. – 704 с.
247. Танаев В. С. Математические модели и методы календарного планирования / В. С. Танаев, Ю. Н. Сотсковым, В. А. Струсевичем. – Мн. : Універсітэцкае, 1994. – 304 с.
248. Танаев В. С. Теория расписаний: Одностадийные системы. / В. С. Танаев, В. С. Гордон, Я. М. Шафранский. – М. : Наука, 1984. – 206 с.
249. Танаев В. С. Введение в теорию расписаний / В. С. Танаев, В. В. Шкурба. – М. : Наука, 1975. – 256 с.
250. Татт У. Теория графов / пер. с англ. / У. Татт. − М. : Мир, 1988. − 424 с.
251. Таха Х. А. Введение в исследование операций : в 2 кн. / Х. А. Таха. – М. : Мир, 1985.
Кн. 1. – 1985. − 480 с.
Кн. 2. – 1985. − 496 с.
252. Теория расписаний и вычислительные машины / под ред. Э. Г. Кофмана. – М. : Наука, 1984. – 334 с.
253. Тимофеева Н. К. Комбинаторные функции в задаче размещения / Н. К. Тимофеева / Компьютерная математика. – 2004. − № 1. – С. 47–56.
254. Тимофеева Н. К. О некоторых свойствах разбиений множества на подмножества / Н. К. Тимофеева // Управляющие системы и машины. – 2002. – № 5. – С. 6–23.
255. Тимофеева Н. К. О способах образования аргумента целевой функции в задачах комбинаторной оптимизации / Н. К. Тимофеева // Кибернетика и ситемный анализ. – 2002. − № 6 – С. 96–103.
256. Тимофеева Н. К. Самонастраивающиеся алгоритмы нахождения неопределенных параметров в задачах комбинаторной оптимизации / Н. К. Тимофеева // Управляющие системы и машины. – 2009. − № 4. – С. 43–51.
257. Тимофеева Н. К. Упорядочение множества значений аргумента целевой функции в комбинаторной оптимизации / Н. К. Тимофеева // Кибернетика и системный анализ. – 1998. – № 6. − С. 78–87.
258. Трухаев Р. И. Модели принятия решений при неопределенности / Р. И. Трухаев. – М. : Наука, 1981. – 258 с.

259. Уилсон Р. Введение в теорию графов / пер. с англ. / Р. Уилсон. − М. : Мир, 1977. – 208 с.
260. Фишберн П. Теория полезности для принятия решений / П. Фишберн. – М. : Наука, 1978. – 352 с.
261. Харари Ф. Перечисление графов / Ф. Харари, Э. Палмер. − Мир, 1977. Diestel R. Graph Theory, Electronic Edition. – NY : Springer-Verlag, 2005. – С. 422.
262. Харари Ф. Теория графов / Ф. Харари. − М. : Мир, 1973. – 300 с.
263. Химические приложения топологии и теории графов / под ред. Р. Кинга / пер. с англ. – М. : Мир, 1987. – 111 с.
264. Холл М. Х. Комбинаторика / М. Х. Холл. − М. : Мир, 1970 – 424 с.
265. Чарин В. С. Линейные преобразования и выпуклые множества / В. С. Чарин. – К. : Вища школа, 1978. – 191 с.
266. Червак Ю. Ю. Оптимізація. Непокращуваний вибір / Ю. Ю. Червак. – Ужгород : Ужгородський національний університет, 2002. – 312 с.
267. Черников С. Н. Линейные неравенства / С. Н. Черников. – М. : Наука, 1968. – 488 с.
268. Шевченко В. Н. Характеристические многочлены многоиндексных транспортных задач // Дискретная математика. – 2003. – Т. 15. – № 2. – С. 83–88
269. Шевченко В. Н. Качественные вопросы целочисленного программирования. – М. : Физматлит, 1995. (English. transl. : Shevchenko V. N. Qualitative topics in integer linear programming. – American Mathematical Society, Providence, Rhode Island, 1997.)
270. Шевченко В. Н. О разбиении выпуклого политопа на симплексы без новых вершин // Известия ВУЗ. Математика. – 1997. – № 12. – С. 89–99.
271. Шевченко В. Н. Дискретный анализ и исследование операций / В. Н. Шевченко, Д. В. Груздев. – Сер. 2. – 2006. – Т. 10. – № 1. – С. 53–64.
272. Шевченко В. Н. Об f-векторах пирамидальных триангуляций точечных конфигураций / В. Н. Шевченко, Д. В. Груздев // Дискретный анализ и исследование операций. – Сер. 2. – 2008. – Т. 15. – № 3. – С. 74–90.
273. Шевченко В. Н. Среднее значение квадрата минора матрицы ограничений аксиальной транспортной задачи

/ В. Н. Шевченко, Е. Б. Титова // Автоматика и телемеханика. – 2004. – № 2. – С. 113–117.
274. Шибанов С. Е. Многокритериальный подход к задаче оптимизации альтернативных маршрутных стратегий в ненадежной сети передачи данных / С. Е. Шибанов, В. П. Шило // Численные методы и технология разработки пакетов прикладных программ. – К. : Ин-т кибернетики им. В. М. Глушкова НАНУ, 1990. – С. 19–23.
275. Шоломов Л. А. Логические методы исследования дискретных моделей выбора / Л. А. Шоломов. – М. : Наука, 1989. – 288 с.
276. Шор Н. З. Квадратичные экстремальные задачи и недифференцируемая оптимизация / Н. З. Шор, С. И. Стеценко. – К. : Наукова домка, 1989. – 208 с.
277. Шор Н. З. Декомпозиционные методы в дробно-линейном программировании / Н. З. Шор, Д. И. Соломон. – Кишенев : Штиинца, 1989. – 204 с.
278. Штойер Р. Многокритериальная оптимизация. Теория, вычисления и приложения / Р. Штойер. – М. : Радио и связь, 1992. – 504 с.
279. Элементы теории геометрического проектирования / С. В. Яковлев, Н. И. Гиль, В. М. Комяк и др.; под ред. В. Л. Рвачева. – К. : Наукова думка, 1995. – 240 с.
280. Яковлев С. В. Оптимизационные задачи синтеза стохастических матриц / С. В. Яковлев // Радиоэлектроника и информатика. – 2005. – № 4. – С. 36–38.
281. Яковлев С. В. Свойства эвклидовых комбинаторных множеств-расстановок с повторениями и без повторений / С. В. Яковлев // Проблемы бионики. – 1989. − № 42. – С. 90–95.
282. Яковлев С. В. О минимизации линейной функции на вершинах перестановочного многогранника с учетом линейных ограничений / С. В. Яковлев, О. А. Валуйская // Доповіді НАН України. – 1999. – № 4. – С. 103–108.
283. Яковлев С. В. О некоторых классах задач оптимизации на множестве размещений и их свойствах / С. В. Яковлев, И. В. Гребенник // Известия вузов. Математика. – 1991. – № 11. – С. 74–86.

284. Aardal K. Polyhedral techniques in combinatorial optimization I: Computations / K. Aardal, S. Hoesel // Statist. Neerlandica. – 1996. – № 15. – P. 3–26.
285. Aardal K. Polyhedral techniques in combinatorial optimization II : Computations / K. Aardal, S. Hoesel // Ibid. – 1999. – Issue 53. − P. 131 – 177.
286. Aardal K., Van Hoesel C.P.M. Polyhedral techniques in combinatorial optimization I: Theory // Statistica Neerlandica. – 1999. – V. 53. – №. 2. – P. 131–177.
287. Arora S. R. Enumeration technique for the set covering problem with a linear fractional functional as its objective function / S. R. Arora, M. C. Puri // ZAMM. – 1977. – Pt. 3,Vol. 57. – P. 181–186.
288. Avis D., Bremer D., Seidel R. // Computational Geometry: Theory and Applications. – 1997. – V. 7. – № 5–6. – P. 265–301.
289. Bauschke H. H., Borwein J. V. On Projection Algorithms for Solving Convex Feasibility Problems // SIAM Revs. – 1996. – 38(3). – P. 367–426.
290. Blum C. An introduction to metaheuristic techniques / C. Blum, A. Roli, E. Alba // Parallel metaheuristics: A new class of algorithms (Ed. E. Alba). – Hoboken: John Wiley & Sons. − 2005. – P. 3–42.
291. Blum C. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison / C. Blum, A. Roli // ACM Computing Surveys. – 2003. – № 3. – P. 268–308.
292. Blum C. An introduction to metaheuristic techniques / C. Blum, A. Roli, E. Alba; ed. E. Alba // Parallel metaheuristics: A new class of algoritms. – Hoboken : John Wiley Sons. – 2005. – 203 p.
293. Blum C. Vetaheuristics in Combinatorial Optimization Overview fnd Conctptual Comparison / C. Blum, A. Roli // ACM Computing Surveys. – 2003. – Vol. 35. – № 3. – P. 268−308.
294. Burkard R. E. A relationship between optimality and efficiency in multichteria 0-1 programming problems / R. E. Burkard, H. Keiding, J. Krapup, P. M. Pruzan // Comput. Oper. Res. − 1981. − V. 8. – № 2. − P. 241–247.

295. Chadha S. S. Dual for a linear fractional program with variable coefficients / S. S. Chadha // Mathematical Reports of Academia Science Canada. – 1995. – Vol. 17. – №. 1. – P. 43–48.
296. Chadha S. S. Duality for a family of nonlinear fractional programming problems / S. S. Chadha, R. A. Heeg // Mathematical Reports of Academia Science Canada. – 1996. – Vol. 18. – №. 6. – P. 237–242.
297. Chinchuluum A. A survey of recent developments in multiobjective optimization / A. Chinchuluum, P. M. Pardalos // Ann. Oper. Res. – 2007. – Vol. 154. – P. 29–50.
298. Christofides N. Graf Theory: An Algorithmic Approach / N. Christofides. – New York; London; San Francisco: Academ. Press, 1975. – 275 p.
299. Calpine H. C. Some properties of Pareto-optimal choices in decision problems / H. C. Calpine, A. Golding // OMEGA. – 1976. − Vol. 4, № 2. − P. 141–147.
300. Dattorro J. Convex Optimization and Euclidean Distance Geometry. Meboo Publishind: Palo Alto, California, 2005.
301. Deriving the number of «good» permutations, with applications to cryptography / C. Cooper, R. Gilchrist, I. N. Kovalenko, D. Novakovic // Кибернетика и системный анализ. – 1999. – № 5. – С. 10–17.
302. Diestel R. Graph Theory, Electronic Edition / R. Diestel. – NY : Springer-Verlag, 2005. – С. 422.
303. Ehrgott M. A survey and annotated bibliography of multiobjective combinatorial optimization / M. Ehrgott, X. Gandibleaux // OR. Spektrum – 2000. – Vol 22. – P. 425–460.
304. Emelichev V. A. The tests of efficiency for a discrete multicriteria optimization problem d’analyse numerique et de theorie de l’approximation / V. Emelichev, O. A. Yanushkevich // Romania. − 1998. − V. 27. − № 2. − P. 237–242.
305. Fukuda K. // Lecture Notes in Computer Science / K. Fukuda, A. Prodon. 1996. – V. 1120. – P. 91–111
306. Gaiha P. Adjacent vertices on a permutohedron / P. Gaiha, S. Gupta // SIAM Journal Appl. Math. – 1977. – Vol. 32. – № 2. – P. 323–327.
307. Grundel D. On the Number of Local Minima for the Multidimensional Assignment Problem / D. Grundel, P. Krokhmal, C. Oliveira, P. Pardalos // J. Of Combinatorial Optimization. – 2007. − № 13. – P. 1–18.

308. Gulati T. R. Efficiency in linear fractional vector maximization problem with nonlinear constraints / T. R. Gulati, M. A. Islam // Optimization. – 1989. – Vol. 20. – № 4. – P. 477–482.
309. Jansson C. Rigorous solution of linear programming problems with uncertain data / C. Jansson, S.M. Rump // ZOR-methods and Models of Oper. Res. – 1991. – P. 87–111.
310. Kearfott R. B. Applications of interval computations: an introsuction / R. B. Kearfott, V. Kreinovich. – Dordrecht : Kluwer, 1996. – P. 1–22.
311. Kelley I. E. The cutting plane method for solving convex programs / I. E. Kelley // SIAM Journal. – 1960. – Vol. 8. – P. 703–712.
312. Kornbluth J. S. H. Multiple objective linear fractional programming / J. S. H. Kornbluth, R. E. Steuer // Management Science. − 1981. – Vol.27, № 9.− P. 1024–1039.
313. Lebedeva T. T. Stability of vector integer optimization problems with quadratic criterion functions / T. T. Lebedeva, N. V. Semenova, T. I. Sergienko // Theory of stochastic processes. – 2004. – № 3–4. – P. 95–101.
314. Malivert C. Structure of efficient sets for strictly quasi convex objectives / C. Malivert, N. Boissard // Journal of Convex Analysis. – 1994. – Vol. 1, № 2. – C.143–150.
315. M. M. Deza, M. Laurent. Geometry of Cuts and Metrics, Springer-Verlag, Berlin, 1997.
316. Motzkin T. S., Raiffa H., Thompson G. L., Thrall R. M. // Contributions to Theory of Games. V. 2. Princeton University Press: 1953.
317. On the Number of Lokal Minima for the Multidimensional Assignmtnt Problem / Grundel D., Krokhmal P., Oliveira C., Pardalos P. // Journal of Combinatorial Optimization. – 2007. – № 13. – P. 1−18.
318. Pacelli G. Optimization of a linear fractional function on a hypersphere of an affine space / G. Pacelli // Journal Optimiz. Theory and Appl. – 1995. – Vol. 84, № 2. – P. 407–414.
319. Papadimitriou C. H. Combinatorial optimization: algorithms and complexity / C. H. Papadimitriou, K. Steiglitz. – [second ed.] – N. Y. : Dover Publications, 1998. – 496 p.
320. Semenova N. Multicriterion problems on the combinatorial set of polyarrangements / N. Semenova, L. Kolechkina // Supple-

ment to International Journal «Information Technologies and Knowledge». – 2009. – Vol. 2. − P. 115–127.
321. Semenova N. V. Vector combinatorial problems in a space of combinations with linear fractional functions of criteria / N. V. Semenova, L. M. Kolechkina, A. M. Nagirna // Information Theories and Applications. – 2008. – Vol. 15. – № 3. – P. 240–245.
322. Smale S. Global analysis and economics, V. Pareto theory with constraints / S. Smale // J. Math. Econ. – 1974. – № 1. – P. 213−221.
323. Steuer R. E. Multiple criteria decision making combined with finance: a categorized bibliography / R. E. Steuer, P. Na // European Journal of Operational Research. – 2003. – Vol. 150. – P. 496–515.
324. Takeda E. Multiple criteria decision problems with fuzzy domination structures / E. Takeda, T. Nishida // Fuzzy sets and syst. – 1980. – Vol. 3. – P. 123–136.
325. Teghem J. Jr. A survey of techniques for finding efficient solutions to multi-objective integer linear programming / J. Jr. Teghem, P. L Kunsch // Asia-Pacific of Operational Research. – 1986. − Vol. 3. − P. 95–108.
326. Toshihide I. Integer programming formulation of combinatorial optimization problems / I. Toshihide // Discrete mathematic. – 1976. − Vol. 16, № 1. − P. 39–52.
327. Veselov S. I., Chirkov A. J. // Discrete Optimization. – 2009. – V. 6. – №. 2. – P. 220–222.
328. Vincke Ph. Analysis of multicriteria decision aid in Europe / Ph. Vincke // European Journal of Operational Research. – 1986. – Vol. 25. – P. 160−168.
329. Ziegler G. M. Lectures on Polytopes. Springer, 2007
330. Yaman H. The robust spanning tree problem with interval data / H. Yaman, O. E. Karasan, M. C. Pinar // Oper. Res. Letters. – 2001. – Vol. 29. – P. 31–40.
331. http://eqworld.ipmnet.ru/ru/library/books/Kirsanov2007ru.pdf
332. http://vuz.exponenta.ru/PDF/book/GrMaple.pdf

Монографія

Донець Георгій Панасович
Колєчкіна Людмила Миколаївна

ЕКСТРЕМАЛЬНІ ЗАДАЧІ НА КОМБІНАТОРНИХ КОНФІГУРАЦІЯХ

Головний редактор М. П. Гречук
Комп'ютерна верстка О. С. Корніліч

Підписано до друку 16.11.2010 р.
Формат 60×84/16. Тираж 300 прим. Ум. друк. арк. 19,3.
Папір 70 г/м^2. Гарнітура «Таймс». Зам. № 230/452

Видавництво
Вищого навчального закладу Укоопспілки
«Полтавський університет економіки і торгівлі»

36014, Полтава-14, вул. Коваля, 3, кімн. 115

Свідоцтво про внесення до Державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції ДК № 3827 від 8.07.2010 р.

Видрукувано з оригінал-макета у редакційно-видавничому відділі видавництва Вищого навчального закладу Укоопспілки
«Полтавський університет економіки і торгівлі»
36014, Полтава-14, вул. Коваля, 3, кімн. 115