https://www.scilook.eu/index.php/slif/article/view/slif15-005
DOI: 10.30888/2415-7538.2019-15-01-005

УДК 519.85

RESEARCH OF DISCRETE OPTIMIZATION PROBLEMS ON COMPUTER

ДОСЛІДЖЕННЯ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ НА КОМП'ЮТЕРІ

Yurchenko I.V. / Юрченко І.В.
c.f.-m.s., as. prof. / к.ф.-м.н., доц.
ORCID: 0000-0001-9929-5758
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Kotsjybynskogo, 12, 58012
Чернівецький національний університет, Чернівці, вул.Коцюбинського, 12, 58012
Blyacher D.V. / Бляхер Д.В.
student / студентка
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Kotsjybynskogo, 12, 58012
Чернівецький національний університет, Чернівці, вул.Коцюбинського, 12, 58012

Анотація. *У роботі розглядаються класичні задачі дискретної оптимізації та їх розв'язання на комп'ютері засобами мови програмування Python.*

Ключові слова: *дискретна оптимізація, задача комівояжера, задача про пакування рюкзака.*

Вступ.

В економічних, виробничих, технологічних процесах різних галузей економіки виникають задачі, подібні за постановкою, що мають ряд спільних ознак та розв'язуються подібними методами. Деякий процес може розвиватися за різними варіантами, кожен з яких має свої переваги та недоліки, причому таких варіантів може бути безліч. Необхідно з усіх можливих варіантів вибрати найкращий. З цією метою використовуються математичні методи. Складність економічних систем (явищ, процесів) як об'єктів досліджень вимагає їх ретельного вивчення з метою з'ясування найважливіших функціональних залежностей, внутрішніх взаємозв'язків між їхніми елементами. В результаті здійснюються можливі спрощення та допущення, що погіршує адекватність побудованих математичних моделей. Не існує загальних рекомендацій щодо процесу моделювання, тому в кожному конкретному разі вимоги до побудови математичної моделі залежать від цілей та умов досліджуваної системи. Одним з класів задач моделювання економічних та інших процесів є задачі дискретної оптимізації.

Основна частина.

Задача дискретної оптимізації – це задача пошуку максимуму або мінімуму функції *f*, визначеної на скінченній або зліченій множині *D* [1]:

$$f\,(x) \to extr\,,\;\; x \in D\,. \tag{1}$$

Функція *f* називається цільовою функцією, елементи множини D – допустимими розв'язками. Якщо множина *D* задається системою обмежень

$$g_i\left(x\right) \le 0, \quad i = 1,\dots,m_1, \quad g_i\left(x\right) = 0, \quad i = m_1 + 1,\dots,m,$$

$$x = \left(x_1,\dots,x_n\right) \in \mathbf{R}^n\,, \qquad x_j \in \Omega_j \subset \mathbf{R}\,, \qquad j = 1,\dots,n_1, \quad n_1 \le n\,,$$

де кожне Ω_j – або скінченна множина, що має не менше ніж два елементи, або злічена множина, тоді задача (1) називається задачею частково дискретного (або дискретного, якщо $n_1 = n$) математичного програмування. Зазвичай передбачається, що параметри задачі – коефіцієнти цільової функції $c_j, j = 1, \ldots, n$, елементи матриці системи обмежень $A = (a_{ij})$ і вектора правих частин $b = b_1, \ldots, b_m$ – цілі числа.

Виділяють [1] наступні основні класи задач дискретного програмування: транспортна задача і її варіанти, задачі з неподільностями, екстремальні комбінаторні задачі, задачі на некласичних областях, задачі з розривною цільовою функцією. *Транспортна задача* лінійного програмування при цілочисельних вихідних даних завжди має цілочисельний оптимальний план. Тому для розв'язання дискретних задач, які можна сформулювати як транспортні задачі, доповнені умовами цілочисельності змінних, можна застосовувати звичайні методи лінійного програмування. Найвідоміша задача цього класу – задача про призначення. *Задачі з неподільностями* – це математичні моделі прикладних задач, змінні в яких є фізично неподільними величинами. Такі задачі описують, наприклад, планування випуску окремих видів продукції або використання неподільних виробничих факторів, задачі розподілу ресурсів і капіталовкладень, мережевого планування і управління, виробничого планування і т.п. У *комбінаторних задачах* оптимізується функція, задана на скінченній множині, елементами якої є вибірки (перестановки) з n об'єктів. З комбінаторних задач, що мають велике прикладне значення, слід виділити *задачу комівояжера* і задачі теорії розкладів. При постановці комбінаторних задач у вигляді задач цілочислового математичного програмування часто вводяться булеві змінні $x_j \in \{0,1\}$, що мають логічний характер: $x_j = 1$, якщо виконується деяка умова і $x_j = 0$ в іншому випадку. *Задачі на некласичних областях* є задачами знаходження екстремуму лінійної функції на неопуклій або незв'язній області, що задається, наприклад, з використанням логічних умов виду «або-або». У *задачах з розривними цільовими функціями*, навпаки, допустима множина – опуклий багатогранник, але цільова функція не є неперервною. До появи подібних цільових функцій призводить, зокрема, облік в моделях постійних витрат, які повинні бути зроблені незалежно від обсягу виробництва. Задачі останніх двох класів можуть бути зведені до задач частково цілочисельного лінійного програмування шляхом введення додаткових, як правило, булевих змінних.

Наведемо приклади постановки деяких задач дискретної оптимізації.

Задача про пакування рюкзака – задача цілочисельного лінійного програмування з одним обмеженням [2]:

$$\sum_{j=1}^{n} c_j x_j \to \max , \tag{2}$$

$$\sum_{j=1}^{n} a_j x_j \le b , \tag{3}$$

$$x_j \ge 0 \in \mathbf{Z}, \quad j = 1, \ldots, n . \tag{4}$$

Числа $c_j, a_j, (j = 1, \ldots, n), b$ – додатні й цілі, $a_j \le b$. Свою назву задача отримала завдяки наступній її інтерпретації. Є n видів неподільних предметів з вартостями $c_1,...,c_n$ і вагами $a_1,...,a_n$. Потрібно так упакувати ними рюкзак, щоб його вага не перевищувала b, а сумарна вартість упакованих предметів була максимальною. Вводячи змінні x_j, $j=1,...,n$ для кількості упакованих предметів кожного типу, отримуємо задачу (2)–(4). Іноді розглядається задача про рюкзак з обмеженням типу рівності $\sum_{j=1}^{n} a_j x_j = b$, що відповідає вимозі повного завантаження рюкзака вантажопідйомністю b. Задача про рюкзак (2)–(4) може бути зведена до задачі з обмеженням типу рівності введенням додаткової невід'ємної цілочисельної змінної $x_{n+1} = b - \sum_{j=1}^{n} a_j x_j$.

Якщо вважати, що є рівно по одному предмету кожного типу, то обмеження (4) слід замінити умовою

$$x_j \in \{0,1\}, \quad j=1,...,n, \tag{5}$$

тобто $x_j = 1$, якщо j-й предмет кладеться в рюкзак, і $x_j = 0$, якщо ні. Отримана задача (2), (3), (5) називається задачею про бінарний або “0-1” рюкзак. При цьому передбачається додатково, що $\sum_{j=1}^{n} a_j > b$, тобто всі предмети в рюкзак упакувати не можна. Множина D в цій задачі – множина n-вимірних бульових векторів з компонентами 0, 1, які відповідають умові (3). Очевидно, що $|D| = 2^n$.

Узагальненням задачі про рюкзак є задача про багатовимірний рюкзак

$$\sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i=1,...,m, \ x_j \in (0,1), \quad j=1,...,n, \tag{6}$$

що має наступну економічну інтерпретацію. Нехай є n проектів, очікуваний прибуток від реалізації яких становить $c_1,....,c_n$. Заданий вектор ресурсів $b = b_1,...,b_m$, $b_i > 0$, $i=1,...,m$; кількість одиниць ресурсу типу i, необхідних для реалізації проекту з номером j, дорівнює $a_{ij} > 0$. Для будь-якого ресурсу i виконується умова $\sum_{j=1}^{n} a_{ij} > b_i$, тобто реалізація всіх проектів неможлива. Потрібно вибрати набір проектів з максимальним сумарним прибутком. Змінна x_j, $j=1,...,n$, набуває значення 1, якщо проект j реалізується і 0 – у протилежному випадку. Множина допустимих розв’язків цієї задачі – це множина булевих векторів $x = (x_1,...,x_n)$, що задовольняють умови (6). Задачі про рюкзак з обмеженнями типу нерівностей завжди мають допустимий розв’язок (наприклад, нульовий). Допустимий розв’язок можна отримати, наприклад, якщо відкинути умову цілочисельності, для задач про бінарний рюкзак – замінити умови (5) обмеженнями $0 \le x_j \le 1$, $j = 1,...,n$, розв'язати отриману задачу лінійного програмування й заокруглити нецілі значення змінних x_j в меншу сторону.

Задача комівояжера. Комівояжер, що знаходиться в рідному місті, повинен відвідати кілька інших населених пунктів, побувавши в кожному по одному разу, і повернутися назад так, щоб загальна довжина його шляху була якомога меншою.

Нехай всі пункти (включаючи початковий) пронумеровані та їх загальна кількість дорівнює *n*. Відстань від пункту *i* до пункту *j* позначається c_{ij}. Вважаючи $c_{ii} = \infty$ (комівояжеру забороняється залишатися на місці), визначаємо $n \times n$ матрицю $C = (c_{ij})$. Природно вважати, що $c_{ij} > 0$.

Будь-який маршрут комівояжера повністю визначається порядком відвіданих пунктів і тому має вигляд $z = (i_1, i_2, ..., i_n, i_1)$. Довжина маршруту *z* – сума відповідних елементів матриці *C*, тобто $l(z) = \sum_{k=1}^{n} c_{i_k, i_{k+1}}$, де $i_{n+1} = i_1$. Позначивши множину допустимих маршрутів через *D*, отримуємо задачу комбінаторної оптимізації $l(z) \to \min$, $z \in D$.

Очевидним практичним застосуванням задачі комівояжера є планування маршрутів перевезення вантажів. Крім цього, розв'язання задачі комівояжера потрібне при розрахунку авіаційних ліній, оптимізації конвеєрного виробництва, складанні розкладів роботи устаткування і т. ін. Під «пунктами» розуміються, наприклад, етапи виробничого процесу або різні операції, що виконуються на одному обладнанні. Отже, c_{ij} може являти собою не тільки відстань, але і час, витрати, інший вимірювач, що служить для визначення «вигідності» маршруту. У загальному випадку, c_{ij} – вартість проходження *j* безпосередньо після *i*.

Нехай $G = (V, A)$ – повний граф з множиною вершин $V = \{1,...,n\}$ і множиною дуг *A*, довжина дуги (*i*,*j*) в якому дорівнює c_{ij}. Контур (орієнтований цикл), що містить кожну вершину графа рівно один раз, називається *контуром Гамільтона*. Таким чином, задача комівояжера – це задача пошуку контуру Гамільтона, що має мінімальну довжину.

В силу замкненості маршруту можна вважати, що початковий пункт i_1 заданий, тому кількість допустимих маршрутів дорівнює (*n* – 1)! Якщо $c_{ji} = c_{ij}$ для всіх $i, j = 1,...,n$ (матриця *C* симетрична), граф *G* є неорієнтованим, і говорять про цикли Гамільтона. Оскільки перестановки $(i_2,...,i_n)$ і $(i_n,...,i_2)$ визначають один маршрут, кількість різних замкнутих маршрутів складе $(n-1)!/2$.

Сформулюємо [1] задачу комівояжера на мові математичного програмування. Нехай $c_{ii} = M$, де *M* – довільне велике число. Уведемо змінні

$x_{ij},\ i,j=1,...,n,$ які набувають значення 1, якщо дуга (i,j) міститься в оптимальному маршруті, і 0 – у протилежному випадку. Тоді сумарна довжина маршруту має вигляд

$$\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}\,. \tag{7}$$

Набору змінних $\{x_{ij}\}$ відповідає граф з n вершинами, дуга (i,j) в якому існує тоді і тільки тоді, якщо $x_{ij}=1$. Зрозуміло, що для допустимого розв'язку задачі цей граф повинен бути циклом, тобто бути зв'язним, і степінь кожної вершини повинна дорівнювати двом. Виконання другої вимоги гарантують умови одноразового відвідування кожного пункту:

$$\sum_{j=1}^{n}x_{ij}=1,\quad i=1,...,n, \tag{8}$$

$$\sum_{i=1}^{n}x_{ij}=1,\quad j=1,...,n, \tag{9}$$

Один з можливих варіантів забезпечення зв'язності маршруту наступний [1]. Для кожної власної підмножини $S\subset V$, $\overline{S}=V\setminus S$, вимагатимемо, щоб $\sum\limits_{i\in S,\,j\in\overline{S}}x_{ij}\geq 1$, тобто повинна існувати принаймні одна дуга, що йде з S в $\overline{S}$. Всього таких обмежень стільки ж, скільки власних підмножин V, тобто 2^n-2.

У більш короткому формулюванні кожній вершині $i=2,...,n$ ставиться у відповідність змінна u_i і слід дотримуватися наступних умов:

$$u_i-u_j+nx_{ij}\leq n-1,\quad i,j=2,...,n,\quad i\neq j\,. \tag{10}$$

Неважко показати, що обмеження (10) виключають наявність циклів довжиною менше n і змінні u_i можна вважати цілочисельними і невід'ємними. Беручи до уваги, що

$$x_{ij}\in\{0,1\},\quad i,j=1,...,n, \tag{11}$$

отримуємо, що задача комівояжера набуває вигляду задачі цілочислового лінійного програмування, в якій є n^2+n-1 змінних і n^2-n+2 обмежень: мінімізувати функцію (7) при обмеженнях (8)–(11).

Існує досить багато різновидів і узагальнень задачі комівояжера [1]. Якщо критерієм оптимальності маршруту комівояжера є не загальна довжина шляху, а протяжність найдовшого переходу, задача комівояжера називається мінімаксною або задачею на вузьке місце. Знімемо тепер обмеження одноразового відвідування кожної вершини. Задача пошуку контуру найменшої довжини, що включає кожну вершину графа хоча б один раз, називається *загальною задачею комівояжера*. Постановка загальної задачі комівояжера зручна, якщо граф не є повним, тобто заборонено безпосередній перехід з деяких пунктів *i* в деякі пункти *j*. Зазвичай відсутнім дугам графа приписується нескінченна вага. Контур Гамільтона в графі може не існувати, тобто задача комівояжера може не мати допустимих рішень. Рішення загальної задачі комівояжера існує, якщо граф сильно зв'язний (будь-які дві вершини можна з'єднати орієнтованим шляхом). У *відкритій задачі комівояжера* повернення в початковий пункт не передбачається: комівояжер починає свій шлях в деякому місті та відвідує кожне місто, що залишилось в точності один раз. При цьому кінцеві пункти маршруту можуть бути довільними або фіксованими. Шлях в графі, що проходить через кожну вершину рівно один раз, називається *шляхом Гамільтона*. Таким чином, відкрита задача комівояжера – це задача пошуку шляху Гамільтона, що має мінімальну довжину. Будь-яку відкриту задачу комівояжера можна звести до звичайної (закритої) або навпаки.

Симплекс-метод [1] — це ітераційна обчислювальна процедура, яка дає змогу, починаючи з певного опорного плану, за скінченну кількість кроків отримати оптимальний план задачі лінійного програмування. Ідея цього методу полягає в здійсненні спрямованого перебору допустимих планів у такий спосіб, що на кожному кроці здійснюється перехід від одного опорного плану до наступного, який за значенням цільової функції був би хоча б не гіршим за попередній. Значення функціонала при переході змінюється в потрібному напрямку [1, 2].

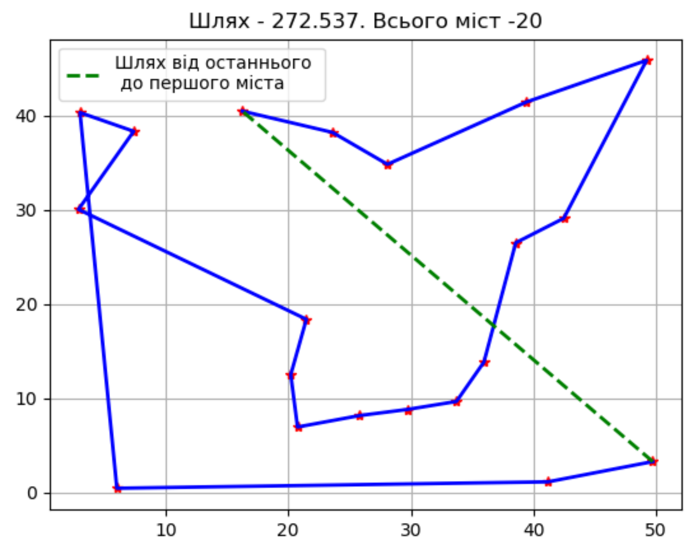
Авторами розроблено програмний продукт на мові Python для розв'язання задачі пакування рюкзака, задачі комівояжера та розв'язання симплекс-методом задачі лінійного програмування. Зокрема, для модельного прикладу

$$\max Z = 8x_1 + 10x_2 - 5x_4$$

$$\begin{cases} 2x_1 + 3x_2 + 4x_3 + 2x_4 \le 450; \\ 3x_1 + 2x_2 + x_3 + 2x_4 \le 380; \end{cases}$$

$$x_j \ge 0;\ j = \overline{1, 4}.$$

після виконання програми реалізації симплекс-методу знайдено оптимальний план задачі $X^* = (x_1 = 48;\ x_2 = 118;\ x_3 = 0;\ x_4 = 0;\ x_5 = 0;\ x_6 = 0)$, $\max Z = 1564$, що збігається з результатами, одержаними в [1, с.77-82].



Для програмної реалізації розв'язання задачі комівояжера методом найближчих сусідів отримано наступні результати (координати міст (x,y) вибираються випадковим чином)

Висновки.

У роботі розглянуті деякі задачі дискретної оптимізації (задача пакування рюкзака, задача комівояжера, симплекс-метод для задачі лінійного програмування). Розроблено програмний продукт на мові Python для розв'язання задачі пакування рюкзака, задачі комівояжера та розв'язання симплекс-методом задачі лінійного програмування.

Література:

1. Наконечний С. І., Савіна С. С. Математичне програмування**:** Навч. посіб.– К.: КНЕУ, 2003.– 452 с.

2. Попов Ю.Д., Тюптя В.І., Шевченко В.І. Методи оптимізації. Навчальний електронний посібник.− Київ: Електр. видання. Ел. бібліотека факультету кібернетики КНУТШ, 2003.− 215 с.

References:

1. Nakonechnyy S.I., Savina S.S. Mathematical programming: Tutorial.– Kyiv: KNEU, 2003.– 452 p.

2. Popov Yu.D., Tjuptja V.I., Shevchenko V.I. Optimization methods. Tutorial.– Kyiv: electr. edition.– Kyiv: Electr. Library of Taras Shevchenko Kyiv National University, 2003.– 215 p.

Abstract. *Some discrete optimization problems (backpack packing problem, traveling salesman problem, simplex method for linear programming problem) are considered in the paper. A software was developed to solve the backpack packing problem, the traveling salesman problem, and the simplex linear problem solving solution with the help of programming language Python.*

Key words: *discrete optimization, traveling salesman task, backpack packing task.*

Стаття відправлена: 10.10.2019 р.