Національна Академія Наук України
Інститут кібернетики імені В.М. Глушкова

На правах рукопису

ДЕЙНЕКО Володимир Григорович

УДК 519.8

Розв'язність в комбінаторній оптимізації:

Розв'язні випадки проблеми комівояжера та евристичні алгоритми

01.05.01 – теоретичні основи інформатики та кібернетики

(математична кібернетика)

АВТОРЕФЕРАТ

дисертації на здобуття наукового ступеня доктора
фізико-математичних наук

КИЇВ 1995

Дисертацією є рукопис.

Робота виконана в Дніпропетровському університеті і частково в Технічному університеті м. Грац (Австрія).

Науковий консультант - профессор Р.Е. Буркад
(Технічний університет м.Грац, Австрія).

Офіційні опоненти:
член-кореспондент НАН України,
доктор фізико-математичних наук,
професор Шор Наум Зуселевич,

член-кореспондент НАН України,
доктор фізико-математичних наук,
професор Анісімов Володимир Владиславович,

академік Академії інженерних наук України,
доктор фізико-математичних наук,
професор Перепелиця Віталій Опанасович.

Провідна установа: Київський технічний університет (КПІ).

Захист відбудеться "13" "жовтня" 1995 р. о 11 годині на засіданні спеціалізованої ради Д 01.39.02 при Інституті кібернетики імені В.М.Глушкова НАН України за адресою:
252022 Київ 22, проспект Академіка Глушкова, 40.

З дисертацією можна ознайомитися в науково-технічному архіві інституту.

Автореферат розісланий "13" "вересня" 1995 р.

Учений секретар
спеціалізованої вченої ради В.Ф. Синявський

Загальна характеристика дисертаційної роботи

Актуальність тематики і ступінь її дослідженості. *Проблема комівояжера* (ПК) може бути сформульована таким чином: надана $n \times n$ матриця відстаней $C = (c_{ij})$, знайти циклічну перестановку π на множині $\{1, 2, \ldots, n\}$, що мінімізує функцію

$$c(\pi) = \sum_{i=1}^{n} c_{i\pi(i)}$$

(комівояжер мусить проїхати всі міста від 1 до n у довільній послідовності і бажає знайти найкоротший шлях). Циклічну перестановку ми будемо називати також туром.

ПК є однією з фундаментальних проблем комбінаторної оптимізації. Ця проблема має важливе теоретичне значення: всі нові методи, що були запропоновані для задач комбінаторної оптимізації, проходили свою першу перевірку на ПК. Ця проблема має також важливе практичне значення, оскільки знаходить своє використання в таких різних сферах, як кристалографія, соціологія, розробка радіоелектронних пристроїв, планування транспортних перевезень, різні проблеми теорії розкладів, тощо.

Відомо, що ця проблема NP–важка. Це означає, що знаходження поліноміальних алгоритмів для розв'язання цієї проблеми малоймовірно. Тому зусилля вчених зосереджені на пошуку ефективних евристичних алгоритмів і пошуку так званих розв'язних випадків, коли проблема може бути розв'язана за поліноміальний час.

Нагадаємо, що проблема мінімізації функції $c(\pi)$ на множині всіх $n!$ перестановок відома як *задача про призначення*. Оптимальне рішення задачі про призначення будемо називати *оптимальним призначенням*. Зрозуміло, якщо оптимальне призначення ϕ є циклічна перестановка, то ϕ є також оптимальне рішення для ПК. Якби ми змогли, наприклад, сформулювати умови, за яких оптимальне призначення є циклічна перестановка, то можна було б казати, що ми описали розв'язний випадок

ПК (оскільки задача про призначення може бути розв'язана за поліноміальний час).

Відомо, що існують тисячі статтей про ПК (класичним оглядом по цій проблемі є книга E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, *The travelling salesman problem*, Wiley, Chichester, 1985), але список розв'язних випадків не такий вже й великий (див. P.C. Gilmore, E.L. Lawler and D.B. Shmoys, Well-solved special cases, Глава 4 у тій же книзі).

Мабуть перший відомий розв'язний випадок ПК – це так званий опуклий випадок: коли всі n точок в площині розташовані на межі їх опуклої оболонки, то оптимальний тур – це обхід вздовж межі.

Цілком природньо забути тепер про геометричні властивості множини точок і проаналізувати комбінаторну структуру матриці відстаней, яка дозволяє знаходити точне рішення в цьому випадку. Цей підхід успішно реалізував Кальмансон (K. Kalmanson, Edgeconvex circuits and the travelling salesman problem, *Canadian Journal of Mathematics* 27, 1975).

Симетрична $n \times n$ матриця C називається матрицею *Кальмансона* ($C \in \mathbb{K}$), якщо виконуються наступні умови:

$$c_{ij} + c_{k\ell} \le c_{ik} + c_{j\ell} \qquad \forall 1 \le i < j < k < \ell \le n \tag{1}$$

$$c_{i\ell} + c_{jk} \le c_{ik} + c_{j\ell} \qquad \forall 1 \le i < j < k < \ell \le n. \tag{2}$$

Кальмансон довів, що для ПК з матрицею Кальмансона тур $\langle 1, 2, 3, \ldots, n \rangle$ є оптимальним туром.

Цікаво, що схожі комбінаторні умови були сформульовані також Супніком (F. Supnick, Extreme hamiltonian lines, *Annals of Math.* 66, 1957). Симетрична $n \times n$ матриця C називається матрицею *Супніка* ($C \in \mathbb{S}$), якщо виконуються наступні умови:

$$c_{ij} + c_{k\ell} \le c_{ik} + c_{j\ell} \qquad \forall 1 \le i < j < k < \ell \le n \tag{3}$$

$$c_{i\ell} + c_{jk} \ge c_{ik} + c_{j\ell} \qquad \forall 1 \le i < j < k < \ell \le n. \tag{4}$$

Супнік довів, що для ПК з матрицею Супніка тур $\langle 1,3,5,7,\ldots,8,6,4,2\rangle$ є оптимальним туром.

Легко бачити, що в умовах Супніка і Кальмансона перший рядок є однаковий. Виявляється, другий рядок в цих умовах можна опустити, і ПК з новою матрицею відстаней залишається поліноміально розв'язною. Для того, щоб сформулювати цей результат точніше, введемо поняття *пірамідального тура*: тур $\phi = \langle 1, i_1, i_2, \ldots, i_r, n, j_1, \ldots, j_{n-r-2}\rangle$ називається *пірамідальним*, якщо $i_1 < i_2 < \cdots < i_r$ і $j_1 > \cdots > j_{n-r-2}$. Хоча кількість пірамідальних турів є експоненціальною, найкоротший тур може бути знайдено за $O(n^2)$ кроків.

Деміденко (В.М. Демиденко, Специальный случай задачи о бродячем торговце, *Изв. Акад. Наук БССР, Сер. физ.-мат. наук* 5, 1976) довів, що для ПК з матрицею відстаней, яка задовільняє умовам (1) (таку матрицю ми будемо називати *симетричною матрицею Деміденка* і визначати це як $C \in \mathbb{D}$), завжди існує оптимальний тур, який є також пірамідальним туром.

Треба відмітити, що стаття Деміденка була не першою, де було впроваджено поняття пірамідального туру. Ця праця була виконана в рамках довгострокових досліджень, які проводились групою Білоруських вчених під керівництвом Д.А. Супруненка.

Відомо, що множина оптимальних рішень ПК має наступні властивості:

(*I*) Множина оптимальних рішень ПК з матрицею відстаней (c_{ij}) є та ж сама, що й для ПК з матрицею відстаней $(c_{ij} + a_i + b_j)$, де (a_i) і (b_i) – два довільні вектори.

(*II*) Довжина оптимального туру ПК з матрицею відстаней (c_{ij}) є та ж сама, що й довжина оптимального туру ПК з матрицею відстаней $(c_{\sigma(i)\sigma(j)})$, тобто перестановка рядків і стовбців матриці відповідно до перестановки σ не змінює довжини оптимального туру. Оптимальний тур для матриці $(c_{\sigma(i)\sigma(j)})$ може бути легко сконструйований з оптимального туру для ПК з матрицею (c_{ij}) перенумерацією елементів відповідно до σ.

Безпосередньо бачимо, що матриці Деміденка, Кальмансона і Супніка зберігають свої комбінаторні властивості після лінійних перетворень. Що ж до перестановки рядків та стовбців, то тут справа складніша. А саме, після перенумерації рядків та стовбців в матриці Деміденка (Кальмансона, Супніка) нова матриця $(c_{\sigma(i)\sigma(j)})$ вже не обов'язково є матриця Деміденка (Кальмансона, Супніка). Це означає, що цілком природньо виникає *проблема розпізнавання*:

> Надана матриця (d_{ij}), знайти перестановку τ таку, що матриця $(d_{\tau(i)\tau(j)})$ є матриця Деміденка (Кальмансона, Супніка).

Якщо така перестновка існує, то матриця (d_{ij}) називається *перевпорядкованою матрицею Деміденка (Кальмансона, Супніка)*, а перестановка τ – *перестановкою Деміденка (Кальмансона, Супніка) для матриці* (d_{ij}).

Фактично, розв'язний випадок є дійсно розв'язним, якщо його можна розпізнати за поліноміальний час. Хоча згадані вище розв'язні випадки відомі вже немалий час (випадок Супніка – з 1957 року, Кальмансона – з 1975, Деміденка – з 1976), відповідні проблеми розпізнавання залишалися відкритими.

Зрозуміло, що для ПК має зміст перевпорядкування рядків та стовбців відповідно до однієї й тієї ж перестановки. Але, припустимо, що існує перестановка σ для рядків і перестановка τ для стовбців такі, що матриця відстаней $(d_{\sigma(i)\tau(j)})$ має спеціальну комбінаторну структуру. Чи не можливо в цьому випадку знайти рішення ПК? Відповідь на це запитання – 'Так', але, звичайно, для матриць із спеціальною комбінаторною структурою – для так званих матриць Монжа.

Матриця C називається матрицею *Монжа*, якщо $c_{ij} + c_{rs} \leq c_{is} + c_{rj}$ для всіх $1 \leq i < r \leq m$, $1 \leq j < s \leq n$. ПК з матрицею відстаней, яка є перевпорядкованою матрицею Монжа, називається ПК Монжа.

Треба відмітити, що для багатьох комбінаторних проблем були знайдені останнім часом нові рішення завдяки чудовим властивостям матриць Монжа. А тому алгоритм розпізнавання перевпорядкованих матриць Монжа має важливе значення не тільки для ПК Монжа.

ПК Монжа є узагальненням відомої ПК Гілморе та Гоморі (P.C. Gilmore and R.E. Gomory, Sequencing a one state variable machine: a solvable case of the travelling salesman problem. *Oper. Research 12*, 1964). Значний внесок в дослідження цієї проблеми був зроблений групою вчених із Мінська під керівництвом Д.А. Супруненка (В.Айзенштат, Н.Гайков, В.Деміденко, Е.Максімович, Н.Метельский, П.Кляус, І.Кунцевіч, Д.Кравчук, В.Сарванов) та вченими із Дніпропетровська (В.Бурдюк і В.Трофімов).

Мета роботи - дослідження відомих розв'язних випадків класичної задачі комівояжера; розробка алгоритмів розпізнавання розв'язних випадків; пошук нових розв'язних випадків і побудова евристичних алгоритмів, які базуються на розв'язних випадках.

Наукова новизна, теоретична та практична цінність роботи. Отримані в дисертації результати містять рішення проблем, які були відомі науковій спільноті як відкриті проблеми напротязі довгого часу. Більш того, для таких проблем, як тур Майстра і проблема оптимальної реалізації алгоритма дерева, були надруковані гіпотези про належність їх до класу NP–важких проблем. В роботі описані також алгоритми, які в деяких випадках узагальнюють добре відомі алгоритми, а в деяких випадках значно скоротчують час, потрібний для знаходження рішення. Обгрунтованість нових теоретичних результатів забезпечується строгістю постановки задач і математичних методів, що застосовуються для їх розв'язання.

Таким чином, у роботі представлено теоретичні положення, сукупність яких можна кваліфікувати як нове вагоме досягнення у дослідженні класичної задачі комбінаторної оптимізації. Нові евристичні алгоритми, що базуються на розв'язних випадках, досліджених в роботі, спроможні змагатися з найкращими евристичними алгоритмами, відомими для ПК. Таким чином, результати дисертації можуть бути також використані і в практичних задачах.

Автор захищає

- Алгоритми розпізнавання перевпорядкованих матриць, які зустрічаються як матриці відстаней у відомих розв'язних випадках ПК, - матриць Монжа, Супніка, Кальмансона, Деміденка.
- Нові розв'язні випадки Евклідової ПК: опукла-оболонка-і-пряма та ПК з Евклідовими матрицями Кальмансона і Деміденка.
- Нові результати в досліджені ПК Монжа, які можуть розглядатися як завершення повного дослідження цього розв'язного випадку ПК:
  - узагальнення відомої стратегії склейки циклів на новий клас матриць;
  - побудова нових ефективних алгоритмів для відомих розв'язних випадків цієї ПК.
- Новий алгоритм динамічного програмування для ПК, який не тільки узагальнює відомі алгоритми для ПК (алгоритм динамічного програмування Хелда і Карпа і алгоритм пошуку оптимального пірамідального туру), але й дозволяє отримати нові теоретичні результати для відомих евристичних алгоритмів для ПК.
- Методику використання розв'язних випадків для побудови евристичних алгоритмів, яка включає:

– побудову нижніх меж, які фактично розширюють множину розв'язних випадків;

– побудову експоненційних околів, оптимальний тур в яких можна знайти за поліноміальний час, і які включають в себе оптимальні тури для відповідних розв'язних випадків.

Апробація результатів.
Основні результати доповідались на Міжнародних симпозіумах з математичного програмування (Амстердам 1991, Анн Арбор 1994), на міжнародних конференціях по Комбінаторній оптимізації (Гронінген 1990, Обервольфах 1991, Оснабрюк 1991, Грац 1993), на Всесоюзних конференціях з проблем теоретичної кібернетики (Горький 1988, Волгоград 1990), на Всесоюзних сімпозіумах з проблем програмного забезпечення рішення задач оптимального планування (Нарва Йиессуу 1978, 1988, Пущино 1980), Всесоюзних семінарах з дискретної математики та її застосуваннях (Москва 1984, 1987), на республіканському семінарі з методів дискретної математики (Судак 1982), на наукових семінарах інституту Кібернетики НАН України (Київ 1988, 1993, 1995), щорічних підсумкових наукових конференціях Дніпропетровського університету (1976 1994) та наукових семінарах кафедри обчислювальної математики і математичної кібернетики (1975 1995).

Публікації. Основні результати дисертації надруковані в роботах [1] [30]. В роботах [24] [28], написаних англійською мовою, повторюються деякі основні результати в переробленому і розширеному вигляді. Роботи [1],[13] та [14] написані разом з науковим керівником кандидатської дисертації Бурдюком В.Я. В цих роботах Бурдюку В.Я. належить постановка задач, розв'язаних автором дисертації. Роботи [2], [9], [15] та [16] написані разом із студентами, які виконували під керівництвом Дейнеки В.Г. курсові та дипломні роботи. Роботи [17] [19] ка [24]–[28] написані у співавторстві з колегами із технічного університету м.Грац (Австрія) та університету м.Гронінген (Нідерланди). В цих роботах автору дисертації належить постановка задач і побудова поліноміальних алгоритмів, для яких пізніше в деяких випадках були запропоновані ефективні реалізації. Ті результати із сумістних статтей, які були вперше отримані іншими співавторами і які згадуються в дисертації, надруковані іншим шрифтом.

Структура та обсяг роботи. Дисертація складається із вступу, семи глав, висновків, літератури (154 назв.), має 192 сторінки (редактор – LATEX, розмір шрифтів – 12*pt*), 39 малюнків і 18 таблиць.

Основні результати дисертації.

(Нижче ми зберігаємо нумерацію глав і розділів дисертації. Нумерація теорем і лем не співпадає з нумерацією їх в дисертації.)

1 Розв'язні випадки Евклідової ПК

В цій главі досліджується Евклідова ПК, де міста – це точки в двумірній площині і відстані вимірюються відповідно до Евклідової метрики. Відомо, що в цьому випадку оптимальний тур не перетинає себе, і таким чином, геометрія наче б то робить проблему трошки легшою. Але і в цьому випадку проблема залишається NP–повною. А це означає, що і для Евклідової ПК дослідження розв'язних випадків є проблема актуальна.

1.1 Опукла оболонка і пряма

Нехай $P = \langle v_1, v_2, \ldots, v_n \rangle \subseteq \mathbb{R}^2$ є послідовність точок в Евклідовій площині і C визначає матрицю відстаней між точками. Якщо матриця відстаней задовільняє умовам Деміденка

(Кальмансона, Супніка), то вона називається *Евклідовою матрицею Деміденка (Кальмансона, Супніка),* а послідовність P називається *послідовністю точок Деміденка (Кальмансона, Супніка)*, і говорять, що точки в P є *множина точок Деміденка (Кальмансона, Супніка).*

Множина точок P називається *опуклою*, якщо всі точки лежать на межі опуклої оболонки. Послідовність точок називається *циклічно впорядкованною*, якщо ці точки утворюють опуклу множину, і точки занумеровані відповідно до розташування на межі опуклої оболонки.

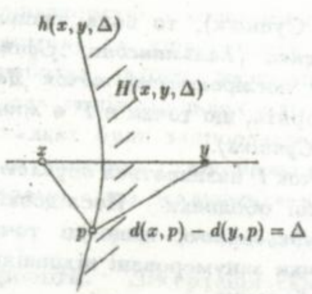
Якщо координати всіх n точок відомі, то циклічне впорядкування може бути знайдене за $O(n \log n)$ елементарних операцій з використанням відомих алгоритмів. Але циклічне впорядкування може буте знайдене також і в тому випадку, коли відома тільки матриця відстаней.

Лема 1. *Для Евклідової матриці відстаней опуклої множини точок P циклічне впорядкування точок може бути знайдено за $O(n \log n)$ елементарних операцій без обчислювання координат точок на площині.*

Наступне цілком природне узагальнення опуклого випадку є так звана ПК *опукла-оболонка-і-пряма*: Евклідова ПК де $n - m$ точок лежать на межі опуклої оболонки n точок, а решта m точок розташовані на відрізку прямої всередині оболонки.

Теорема 2. *ПК опукла-оболонка-і-пряма може бути розв'язана за $O(nm)$ елементарних операцій.*

Легко бачити, що ПК опукла-оболонка-і-пряма є узагальненням ПК на трьох паралельних прямих, яка була досліджена Катлером (M. Cutler, Efficient special case algorithms for the N-line planar traveling salesman problem, Networks 10, 1980). Алгоритм Катлера потребує $O(n^3)$ елементарних операцій.



Мал. 1: Ілюстрація до геометричної інтерпретації

1.2 Геометрична інтерпретація і алгоритми розпізнавання для Евклідових матриць Деміденка і Кальмансона

Для $x, y \in \mathbb{R}^2$ і $\Delta \in \mathbb{R}$ позначимо через $h(x,y,\Delta) = \{p \in \mathbb{R}^2 \mid d(x,p)-d(y,p) = \Delta\}$ множину точок $p \in \mathbb{R}$, які розташовані на гілці гіперболи з фокусними точками x і y, а через $H(x,y,\Delta) = \{p \in \mathbb{R}^2 \mid d(x,p)-d(y,p) \geq \Delta\}$ – множину точок $p \in \mathbb{R}^2$ в нескінченній області, яка обмежена гілкою $h(x,y,\Delta)$ і яка не включає фокусну точку x (див. Мал. 1). Нарешті, визначимо $\Delta_i^k = d(v_{k-1}, v_i) - d(v_k, v_i)$ для $2 \leq k \leq n$.

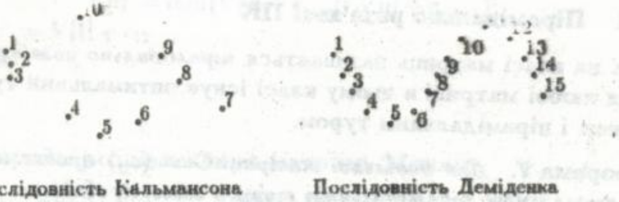
Теорема 3. *Послідовність точок $P = \langle v_1, \ldots, v_n \rangle$ є послідовністю точок Деміденка, якщо і тільки якщо для кожного p, $4 \leq p \leq n$, точка v_p знаходиться в області*

$$H_p = \bigcap_{k=3}^{p-1} H(v_{k-1}, v_k, \Delta^k),$$

де $\Delta^k = \max\{\Delta_i^k | i = 1, \ldots, k-2\}$.

Теорема 4. *Послідовність точок $P = \langle v_1, \ldots, v_n \rangle$ є послідовністю точок Кальмансона, якщо і тільки якщо вона є послідовністю Деміденка і кожна точка $v_p \in P$, $p \geq 4$, знаходиться в області*

$$H'_p = \bigcap_{k=3}^{p-1} \bigcap_{i=1}^{k-2} H(v_{i+1}, v_i, -\Delta_k^{i+1}).$$



Мал. 2: Розв'язні випадки Евклідової ПК

Сформульовані теореми дозволяють будувати послідовності Деміденка і Кальмансона на площині. На мал. 2 зображені дві такі послідовності. Тур $(1, 2, 3, 4, 5, 6, 7, 8, 15, 14, 13, 12, 10, 9)$ є оптимальним туром для ПК на зображеній множині точок Деміденка.

Геометрична інтерпретація дозволила краще зрозуміти природу досліджуваних проблем і знайти алгоритми розпізнавання відповідних матриць відстаней. Розділ 1.3 дисертації описує алгоритм розпізнавання Евклідової матриці Кальмансона.

Теорема 5. *Для $n \times n$ Евклідової матриці відстаней C за $O(n^2)$ елементарних операцій можна розпізнати, чи є C перевпорядкованою матрицею Кальмансона.*

Розділ 1.4 дисертації описує алгоритм розпізнавання Евклідової матриці Деміденка.

Теорема 6. *Для $n \times n$ Евклідової матриці відстаней C за $O(n^4)$ елементарних операцій можна розпізнати, чи є C перевпорядкованою матрицею Деміденка.*

2 Алгоритми розпізнавання пірамідально розв'язних ПК

2.1 Пірамідально розв'язні ПК

ПК на класі матриць називається *пірамідально розв'язною*, якщо для любої матриці в цьому класі існує оптимальний тур, який є також і пірамідальним туром.

Теорема 7. *Для довільної матриці $C = (c_{ij})$ проблема побудови оптимального пірамідального туру, а саме, проблема*

$$\min\{\sum_{i=1}^{n} c_{i\rho(i)} \mid \rho \text{ - пірамідальний тур}\}$$

може бути вирішена за $O(n^2)$ елементарних операцій.

Як нам відомо, першою роботою, де досліджувались пірамідальні цикли, була стаття В.С.Айзенштата і Д.Н.Кравчука (Алгоритмы отыскания линейной формы на множестве всех полных циклов в некоторых случаях, *Докл. Акад. Наук БССР 12*, 1968). В цій роботі розглядалась ПК з так званою *матрицею добутків* $C = (c_{ij}) = (u_i \cdot v_j)$ з $u_1 \leq u_2 \leq \ldots \leq u_n$ і $v_1 \geq v_2 \geq \ldots \geq v_n$. Пізніше Деміденко (В.Демиденко, Специальный случай задачи о бродячем торговце, *Изв. Акад. Наук БССР, Сер. физ.-мат. наук* 5, 1976) відмітив, що оптимальний пірамідальний тур може бути легко знайдений для кожної матриці. Незалежно від цієї роботи Дейнеко [3] теж запропонував поліноміальний алгоритм, який використовував графічну інтерпретацію пірамідального туру. Ми подаємо тут $O(n^2)$–алгоритм для симетричного випадку із [9], який використовує тільки $O(n)$ пам'яті.

Алгоритм LengthPyr
Input: Матриця відстаней $C = (c_{ij})$.
Output: Довжина c^* найкоротшого туру.
BEGIN
  for $i := 1$ **to** $n-2$ **do**

$$
\begin{array}{l}
\quad V[i] := c_{in} + c_{n-1,n}; \\
\quad \textbf{for } j := n-2 \textbf{ downto } 2 \textbf{ do} \\
\quad\quad \textbf{for } i := 1 \textbf{ to } j-1 \textbf{ do} \\
\quad\quad\quad V[i] := \min\{V[j] + c_{i,j+1};\ V[i] + c_{j,j+1}\}; \\
\quad c^* := V[1] + c_{12} \\
\textbf{END.}
\end{array}
$$

2.2 Алгоритм розпізнавання матриць Монжа

Формально проблема розпізнавання матриць Монжа формулюється таким чином:

> Надана $(n \times n)$ матриця $C = (c_{ij})$, треба розпізнати, чи існують перестановка рядків σ і перестановка стовбців ρ такі, що матриця $(c_{\sigma(i)\rho(j)})$ є матрицею Монжа і, якщо такі перестановки існують, то знайти їх.

Теорема 8. *Проблема розпізнавання перевпорядкованих матриць Монжа може бути вирішена за $O(n^2)$ елементарних операцій.*

Алгоритм для розпізнавання перевпорядкованих матриць Монжа будемо називати Monge-Recognition. Цей алгоритм може також бути використаний для розпізнання перевпорядкованих *контр-Монж матриць*. Матриця C називається *контр-Монж* матрицею, якщо нерівність $c_{ij} + c_{rs} \geq c_{is} + c_{rj}$ виконується для $1 \leq i < r \leq m$ і $1 \leq j < s \leq n$.

Буркард та інші (R.E. Burkard, B. Klinz and R. Rudolf, Perspectives of Monge Properties in Optimization, to appear in Discrete Mathematics) узагальнили алгоритм Monge-Recognition для розпізнавання алгебраїчних матриць Монжа. А Рудольф (Recognition of d-dimentional Monge arrays, *Discrete Applied Mathematics* **52**, 1994) використав цей алгоритм для розпізнавання перевпорядкованих масивів Монжа.

2.3 Тур Майстра: розпізнавання матриць Кальмансона

У відповідньому розділі дисертації розглядається проблема розпізнавання перевпорядкованих матриць Кальмансона.

Теорема 9. *Для симетричної $n \times n$ матриці C за $O(n^2 \log n)$ елементарних операцій можна розпізнати, чи є матриця C перевпорядкованою матрицею Кальмансона.*

Мабуть найцікавішим прикладом використання матриць Кальмансона є проблема пошуку тура Майстра. *Тур Майстра* π для множини точок V – це тур, який має такі властивості: для довільної підмножини $V' \subseteq V$ оптимальний тур на V' отримується із π відкиданням точок, які не належать V'.

Проблема пошуку тура Майстра була сформульована Пападімітріоу на літній школі по комбінаторній оптимізації в 1993 році (Lecture at the Maastricht Summerschool on Combinatorial Optimization, August 1993, див. також книгу C.H. Papadimitriou, Computational Complexity, *Addison-Wesley*, 1994).

Теорема 10. *Для $n \times n$ симетричної матриці відстаней C перестановка $\langle 1, 2, \ldots, n \rangle$ є туром Майстра, якщо і тільки якщо C є матрицею Кальмансона.*

Висновок 11. *Для симетричної $n \times n$ матриці C за $O(n^2 \log n)$ елементарних операцій можна розпізнати, чи існує для неї тур Майстра.*

2.4 Загальний підхід, що дозволяє запобігти заборонені підматриці: розпізнавання матриць Супніка

Багато проблем розпізнавання перевпорядкованих матриць можуть бути сформульовані як проблеми пошуку таких перевпорядкованих матриць, які б запобігали заборонені підматриці. Наприклад, матриця Монжа може бути визначена як матриця, яка не включає заборонені 2×2 підматриці такі, що

$$a_{11} + a_{22} > a_{12} + a_{21}. \qquad (5)$$

У відповідньому розділі дисертації пропонується загальний підхід для розв'язання цієї проблеми. Досліджуються випадки, коли ця проблема є NP–повною, і випадки, коли існують поліноміальні алгоритми. Один із прикладів використання загальної схеми – алгоритм розпізнавання перевпорядкованих матриць Супніка.

Теорема 12. *Перевпорядкована $n \times n$ матриця Супніка може бути розпізнана за $O(n^4)$ елементарних операцій.*

Інший приклад, де може бути використана загальна схема, це розпізнавання неповних матриць Монжа.

Матриця C з деякими невизначеними елементами називається *неповною матрицею Монжа*, якщо для всіх $1 \leq i \leq r \leq n$, $1 \leq j \leq s \leq n$ виконуються наступні умови. Якщо всі елементи c_{ij}, c_{rs}, c_{is} і c_{rj} відомі, то має місце нерівність Монжа $c_{ij} + c_{rs} \leq c_{rj} + c_{is}$.

Теорема 13. *Проблема розпізнавання неповних матриць Монжа є NP–повною.*

З теореми витікає, що розпізнавання неповних матриць Монжа в гіршому випадку потребує експоненціального часу. В дисертації обгрунтовані випадки, коли ця проблема може бути вирішена за поліноміальний час.

3 ПК з перевпорядкованими матрицями Монжа (ПК Монжа)

3.1 Стратегія склеювання циклів: узагальнення результатів Гілморе і Гоморі

Припустимо, що існує перестановка ϕ така, що матриця $c_{i,\phi(j)}$ є матрицею Монжа. В цьому випадку ϕ є оптимальним призначенням для задачі про призначення з матрицею (c_{ij}). Припустимо далі, що ϕ має більш ніж один цикл. Цілком природньо спробувати відшукати оптимальний тур серед турів, які отримані з ϕ об'єднанням в останньому циклів. Ця стратегія отримала назву *склейка циклів*, і може бути сформульована таким чином:

Надане оптимальне призначення ϕ, знайти α^* таку, що $\phi \circ \alpha^*$ є оптимальним туром.

Однією з перших статтей, де досліджуються розв'язні випадки ПК, була згадана вище стаття Гілморе і Гоморі. В цій статті розглянута ПК з матрицею відстаней, елементи якої обчислюються як

$$g_{ij} = \begin{cases} \int_{a_i}^{b_j} h_1(x)dx, & b_j \geq a_i, \\ \int_{b_j}^{a_i} h_2(x)dx, & b_j < a_i. \end{cases}$$

Матрицю (g_{ij}) будемо називати *матрицею Гілморе-Гоморі*, а відповідну ПК - ПК Гілморе–Гоморі. Гілморе і Гоморі довели, що ПК Гілморе–Гоморі може бути розв'язана за $O(n \log n)$ елементарних операцій. Їх алгоритм – це реалізація схеми склейки циклів.

Пізніше В. Бурдюк і В. Трофімов (Обобщение результатов Гилморе и Гомори по решению задачи о коммивояжере, *Изв. АН СССР, Техническая кибернетика 3*, 1970) узагальнили результати Гілморе і Гоморі для матриць Монжа. Незалежно від них В. Сарванов (К минимизации линейной формы на множестве n-членных циклов, *Изв. АН БССР. Сер. физ.-мат. наук* 4, 1976) довів аналогічні результати для ПК з матрицею добутків.

Стратегія склейки циклів використовує слідуючий відомий факт. Нехай перестановка ϕ має два підцикли ϕ_1 і ϕ_2, де $i \in \phi_1$ і $j \in \phi_2$. Перемножуючи перестановку ϕ і транспозицію $\langle i, j \rangle$ ми отримаємо перестановку, де підцикли ϕ_1 і ϕ_2 об'єднані вже в один цикл.

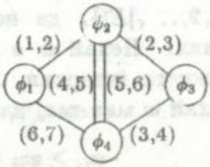
Перестановка α називається *перестановкою склейки*, якщо $\phi \circ \alpha$ є циклічна перестановка. Якщо $\phi \circ \alpha$ є оптимальна перестановка, то α називається оптимальною перестановкою склейки.

Для того, щоб визначити множину перестановок склейки, введемо допоміжний *граф склейки* G_ϕ.

Нехай ϕ має $m(m > 1)$ підциклів ϕ_k, а саме, $\phi = \phi_1\phi_2\ldots\phi_m$ (інакше ϕ вже є оптимальним туром). *Граф склейки* $G_\phi = (V, E)$ будується таким чином. Кожна вершина $v \in V$ відповідає підциклу ϕ. Кожне ребро в E відповідає *суміжній* транспозиції $\langle i, i+1\rangle$, а саме, якщо індекс i належить підциклу ϕ_j (який відповідає вершині ϕ_j) і індекс $(i+1)$ належить підциклу ϕ_k (який відповідає вершині ϕ_k), то дві вершини ϕ_j і ϕ_k з'єднуються ребром $(i, i+1)$.

Граф склейки G_ϕ для перестановки

$$\phi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 5 & 3 & 6 & 2 & 4 & 1 \end{pmatrix}$$
$$= \langle 1,7\rangle\,\langle 2,5\rangle\,\langle 3\rangle\,\langle 4,6\rangle$$



Твердження 14 *(Гілморе і Гоморі). Граф G_ϕ є Ейлерів мультиграф з не більш як $(n-1)$ ребрами.*

Перестановка, яка отримана як добуток суміжних транспозицій, називається *перестановкою дерева*, якщо ребра, що відповідають транспозиціям, формують остове дерево в графі склейки G_ϕ.

Теорема 15 *(Гілморе і Гоморі). Кожна перестановка дерева є перестановкою склейки.*

Теорема 16 *(Бурдюк і Трофімов). Нехай $(s_{i\tau(j)})$ є матриця Монжа. Для кожної циклічної перестановки ϕ існує остове дерево $T = \{(i_1, i_1+1), \ldots, (i_{m-1}, i_{m-1}+1)\}$ в графі G_τ і послідовність σ для перемножування транспозицій, які відповідають T, така що перестановка $\tau_T = \tau \circ \langle i_{\sigma(1)}, i_{\sigma(1)}+1\rangle \circ \ldots \circ \langle i_{\sigma(m-1)}, i_{\sigma(m-1)}+1\rangle$ є циклічною перестановкою і $c(\tau_T) \le c(\phi)$.*

Зрозуміло, теорема 16 є дуже корисною, оскільки вона звужує пошук оптимальної перестановки склейки. Але, як і раніше, проблема залишається NP–повною (див. [28] і В. Сарвалов, О сложности минимизации линейной формы на множестве циклических подстановок, *Докл. АН СССР 253*, 1980).

Дослідження властивостей ПК Монжа дозволили нам звести цю проблему до наступної *проблеми мінімального остового дерева з гілками* (B-MST-проблеми):

> Нехай $G = (V, E)$ є Ейлерівий мультиграф з $E = \{1, 2, \ldots, |E|\}$, де послідоність ребер $1, \ldots, |E|$ формує шлях. Нехай w є вагова функція, яка призначає для кожного інтервалу $[i, \ldots, j]$, $1 \leq i \leq j \leq |E|$ ціле w_{ij}, і нехай w має такі властивості:
>
> $$w_{ij} \geq w_{ik} + w_{k+1,j} \qquad \forall i \leq k < j. \tag{6}$$
>
> Інтервал $[i, \ldots, j]$ називається *гілкою* відносно до довільної множини $E' \subseteq E$, якщо це є максимальний підінтервал E' (а саме, $[i, \ldots, j] \subseteq E'$, $i-1 \notin E'$ і $j+1 \notin E'$). Нехай для $E' \subseteq E$ $Br(E')$ означає множину всіх гілок відносно до E'. *Вага гілок* (або b–*вага*) множини $E' \subseteq E$ визначається як
>
> $$w(E') = \sum_{[i,\ldots,j] \in Br(E')} w_{ij}.$$
>
> *Вага гілок* (b–*вага*) остового дерева $T = (V, E_T)$ в G є $w(E_T)$. B-MST-проблема – це проблема конструювання остового дерева в G з мінімальною b–вагою.

Таке переформулювання проблеми дозволяє розпізнати нові розв'язні випадки ПК, які досліджуються в розділах 3.3 і 3.4 дисертації. Окрім того, нам вдалося знайти нові узагальнення стратегії склеювання циклів.

1. Новий клас матриць. Твердження теореми 16 справедливі для нового класу матриць, який вміщує в себе клас матриць

Монжа, а саме для перевпорядкованих матриць D, для елементів яких виконуються наступні співвідношення:

$$d_{i,i+1} + d_{i+1,i} + d_{ii} + d_{i+1,i+1} \geq 0 \tag{7}$$

$$d_{kj} + d_{ji} - d_{jj} - d_{ki} \geq 0 \quad \text{для } i < j < k \tag{8}$$

$$d_{ij} + d_{jk} - d_{ik} - d_{jj} \geq 0 \quad \text{для } i < j < k \tag{9}$$

$$\text{для } i, i+1, j, k \in \{1, 2, \ldots, n\}.$$

Матриця $D = (d_{ij})$ називається *слабою матрицею Монжа*, якщо для $i < j, l$ мають місце співвідношення

$$d_{ii} + d_{jl} \leq d_{il} + d_{ji}.$$

Твердження 17.

Якщо матриця D має властивості (7)–(9), то

- *D є матриця Демиденка;*
- *D є слабка матриця Монжа.*

Приклад 1. *Матриця* $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ *не є матрицею Монжа, але вона належить до нового класу матриць.*

2. Проблема не більш як p комівояжерів. *Проблема не більш як p комівояжерів* формулюється так. Надана $n \times n$ матриця відстаней $C = (c_{ij})$, знайти перестановку π на множині $\{1, 2, \ldots, n\}$, яка має не більш ніж p циклів, і мінімізує функцію

$$c(\pi) = \sum_{i=1}^{n} c_{i\pi(i)}.$$

В статті [14] вперше було відмічено, що теорема 16 може бути також переформульована для проблеми не більш як p комівояжерів. Різниця полягає тільки в тому, що для довільної перестановки π з не більш як p циклами існує ліс F в G_ϕ (а не дерево) з $(m-p)$ ребрами (m – кількість циклів в оптимальному призначенні ϕ). Використовуючи цей ліс, може бути сконструйована така перестановка α, що $\phi \circ \alpha$ має рівно p циклів і $c(\phi \circ \alpha) \leq c(\pi)$.

Всі розв'язні випадки, які відомі для ПК з матрицею Монжа, мають місце і для проблеми не більш як p комівояжерів (див. [14], [13], [4]).

3.2 Евристичні алгоритми для ПК Монжа: асимптотична поведінка алгоритма Subtour-patch

Алгоритм Subtour-patch. Оскільки ПК Монжа є NP–повною, цілком природньо розглянути евристичний алгоритм для вирішення цієї проблеми.

Властивості функції w суттєво використовуються в алгоритмі, який є простим узагальненням алгоритма Гілморе і Гоморі. Визначемо вагу $p(i)$ ребра $(i, i+1)$ в G_ϕ як $p(i) = w_{ii}$. Нехай $P(T)$ означає вагу остового дерева T в G_ϕ, а саме, суму ваг ребер в T. Якщо T_{min} є мінімальне остове дерево в G_ϕ, то $w(T) \geq P(T_{min})$ для довільного дерева T. Нехай $\tau_{patch} := \phi \circ \alpha_{T_{min}}$ є перестановка склейки, що відповідає T_{min}. Нехай τ_{opt} є оптимальна перестановка. Тоді мають місце наступні нерівності:

$$c(\phi) + P(T_{min}) \leq c(\tau_{opt}) \leq c(\tau_{patch}) = c(\phi) + w(T_{min}).$$

Алгоритм Subtour-patch

Input: Перевпорядкована матриця Монжа.

Output: Циклічна перестановка τ.

BEGIN

Step 1: Розпізнати матрицю Монжа, знаходячи одночасно оптимальне призначення ϕ і перевпорядковуючи матрицю відстаней.
Якщо ϕ є тур, то $\tau := \phi$ є оптимальний тур. Стоп.

Step 2: Побудувати граф склейки G_ϕ.

Step 3: Знайти мінімальне остове дерево T_{min} в G_ϕ.

Step 4: Побудувати перестановку склейки $\alpha_{T_{min}}$, відшукуючи для кожної гілки в T_{min} оптимальний пірамідальний тур.

Step 5: Побудувати тур $\tau_{patch} = \phi \circ \alpha_{T_{min}}$.
Якщо $w(T_{min})(:= c(\tau_{patch}) - c(\phi)) = P(T_{min})$, то τ_{patch} є оптимальний тур.
Інакше, отримано рішення з точністю
$\epsilon = (w(T_{min}) - P(T_{min}))/(c(\phi) + P(T_{min}))$.

END.

ПК з матрицею добутків.
Першою роботою, де досліджувалась ПК з матрицею добутків, була стаття Д. Супруненка (О значениях линейной формы на множествах подстановок, *Кибернетика* 2, 1968). Пізніше багато робіт білоруських вчених було присвячено цій проблемі. ПК з матрицею добутків теж є NP–повною. Але в цьому випадку алгоритм Subtour-patch може бути використаний як дуже добра евристика, про що свідчить слідуюча

Теорема 18. *Нехай* $a_1 \geq 0$, $b_{\phi(n)} \geq 0$ *і* $u_i := a_{i+1} - a_i \geq d > 0$, $v_i := b_{\phi(i)} - b_{\phi(i+1)} \geq d > 0$ *для* $i = 1, 2, \ldots, n-1$. *Нехай далі*

$$M := \max_i \{u_i \cdot v_{i+1}, u_{i+1} \cdot v_i\}.$$

Тоді

$$\frac{c(\tau_{patch}) - c(\tau_{opt})}{c(\tau_{opt})} \leq \frac{6M}{d^2(n-2)^2}.$$

3.3 Розв'язні випадки ПК Монжа: спеціальні матриці

ПК Гілморе-Гоморі. У відповідному розділі дисертації доведено, що алгоритм Subtour-patch знаходить точне рішення для ПК Гілморе-Гоморі. Для цього достатньо було б довести, що $w(T_{min}) = P(T_{min})$ для мінімального остового дерева T_{min} в G_ϕ. Але для ПК Гілморе-Гоморі має місце навіть сильніше твердження:

Теорема 19. *Якщо C є матрицею Гілморе-Гоморі, то $w(T) = P(T)$ для кожного остового дерева в G_ϕ.*

Із теореми негайно витікає, що ПК Гілморе-Гоморі є тільки підклас проблем, які розпізнаються і вирішуються точно алгоритмом Subtour-patch. Це означає, що алгоритм розпізнавання матриць Гілморе-Гоморі (див. R. Chandrasekaran, Recognition of Gilmore-Gomory traveling salesman problem, *Discrete Applied Mathematics* 14, 1986) стає непотрібним, якщо його використовувати тільки для розв'язання ПК Гілморе-Гоморі.

Узагальнення Малої ПК. Розглянемо тепер ПК з матрицею відстаней (c_{ij}) з $c_{ij} = -g_{ij}$, де (g_{ij}) є матриця Гілморе і Гоморі. Ця проблема є узагальненням так званої Малої ПК з матрицею відстаней $C' = (c'_{ij})$ з $c'_{ij} = \min\{a_i, b_j\}$, де (a_i) і (b_i) є два довільні вектори. Відомо, що Мала ПК може бути розв'язана за $O(n)$ елементарних операцій (див. Е.Я. Габович, Малая задача коммивояжера, *Труды выч. центра Тартуск. университета* 19, 1970). Наступна теорема узагальнює результати Габовіча і характеризує структуру графа G_ϕ для ПК Монжа з матрицею відстаней $(-g_{ij})$.

Теорема 20. *Нехай G_ϕ є граф склейки для ПК з матрицею відстаней $(-g_{ij})$, вектор (p_i) – вектор ваг ребер і (w_{ij}) – матриця b-ваг. Тоді*

- *існує не більше одного ребра $(i', i'+1)$ з ненульовою вагою $p_{i'} > 0$;*
- *існує не більше однієї гілки $\{(j, j+1), (j+1, j+2)\}$ з вагою $w_{j,j+1} > w_{jj} + w_{j+1,j+1}$, і якщо така гілка існує, то*
  - *якщо існує i', то $i' \in \{j, j+1\}$,*
  - *тільки гілки, які включають $\{(j, j+1), (j+1, j+2)\}$, мають ненульову b-вагу, яка дорівнює в цьому випадку $w_{j,j+1}$.*

В дисертації доведено, що алгоритм Subtour-patch не знаходить точного рішення для цієї ПК. Але якщо модіфікувати алгоритм

Subtour-patch, то новий алгоритм буде вирішувати точно як ПК Гілморе-Гоморі, так і Малу ПК. Для цього достатньо добавити в алгоритм такий крок:

Алгоритм Subtour-patch: вставка.

Step 3.1: Знайти в T_{min} гілку $\{(i^*-1,i^*),(i^*,i^*+1)\}$ з найбільшою b-вагою w_{i^*-1,i^*}. Якщо $w_{i^*-1,i^*} > 0$, то знайти дерево T без ребра (i^*-1,i^*) чи (i^*,i^*+1); якщо таке дерево знайдене, то замінити $T_{min} := T$.

3.4 Розв'язні випадки ПК Монжа: спеціальний граф склейки.

Історична справка. В багатьох статтях (Айзенштат і Кравчук - 1968, Айзенштат - 1975, Айзенштат і Максимовіч - 1978, Деміденко і Мецельский - 1974, Кунцевіч - 1968) розглядались розв'язні випадки ПК з матрицею добутків. Використовуючи запропоновану нами термінологію, всі ці випадки можуть бути охарактеризовані як випадки із спеціальними графами склейки. А саме, граф склейки в усіх цих випадках був мультишляхом із спеціальною структурою. Сарванов і, незалежно від нього, Дейнеко [4] використали вперше термінологію графа склейки для характеризації розв'язних випадків ПК із спеціальними матрицями. Для випадків, коли граф склейки є мультишлях з довільною структурою, були знайдені алгоритми складності $O(n^3)$. Пізніше Гайков покращив алгоритм Сарванова до $O(mn)$, де m це кількість циклів в оптимальному призначенні. Використовуючи результат Парка для ПК з матрицею Монжа (J.K. Park, A special case of the n-vertex traveling salesman problem that can be solved in $O(n)$ time, *Information Processing Letters* 40, 1991), Буркард і Дейнеко [18] побудували алгоритм складності $O(n)$ для цього спеціального випадку ПК Монжа.

Спеціальний граф склейки: мультишлях.

Нехай $A = (a_{ij})$ є матриця відстаней для довільної ПК, а $\phi = \phi_1\phi_2\ldots\phi_m$ – оптимальне призначення з підциклами $\phi_1, \phi_2, \ldots, \phi_m$. Розглянемо окіл $Q(\phi)$ перестановки ϕ, який вміщує всі циклічні перестановки, що можуть бути представлені як добуток ϕ і транспозицій $\langle i_1, j_2\rangle, \langle i_2, i_3\rangle, \ldots, \langle i_{m-1}, j_m\rangle$, де $i_{k-1} \in \varphi_{k-1}, j_k \in \phi_k$ для $k = 2, 3, \ldots, m$. Відмітимо, що послідовність, в якій розглядаються підцикли, фіксована. Таким чином, ми не отримаємо всіх циклічних перестановок, але для випадку, коли A є матрицею Монжа, а граф склейки – мультишляхом, окіл $Q(\phi)$ вміщує оптимальний тур.

Окіл $Q(\phi)$ був вперше описаний в [9], а в [7] він використовувався в евристичному алгоритмі для ПК. Буркард і Дейнеко [18] використували цей окіл для побудови ефективного алгоритму для ПК Монжа.

Яка перестановка із $Q(\phi)$ вибирається, залежить від того, яка множина транспозицій вибрана і в якій послідовності транспозиції перемножуються з ϕ.

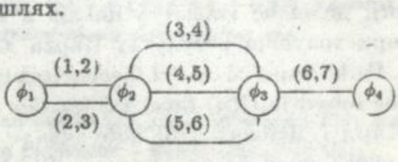
Число перестановок в $Q(\phi)$ обчислюється за формулою

$$|Q(\phi)| = |\phi_1| \cdot |\phi_2| \cdot (|\phi_2| + 1) \cdot \ldots |\phi_{m-1}| \cdot (|\phi_{m-1}| + 1) \cdot |\phi_m| .$$

Теорема 21. ***Найкоротший тур в околі $Q(\phi)$ може бути знайдений за $O(n^2 |\phi|^*)$ елементарних операцій, де $|\phi|^* = \max_{i=1..m} |\phi_i|$.***

Нехай тепер $A = (a_{ij})$ є перевпорядкована матриця Монжа, а граф склейки G_ϕ є мультишлях.

Мультішлях G_ϕ
для
$\phi = \langle 2\rangle\langle 1,3,5\rangle\langle 4,6\rangle\langle 7\rangle$



Оскільки A тепер є матрицею Монжа, достатньо розглядати тільки суміжні транспозиції. Більш того, тепер можна використати результати Царка для ПК Монжа.

Теорема 22. *Якщо G_ϕ для ПК Монжа є мультишлях, то оптимальний тур може бути знайдений за $O(n)$ елементарних операцій.*

Висновок 23. *Якщо G_φ для ПК Монжа є мультицикл, то оптимальний тур може бути знайдений за $O(nm)$ елементарних операцій.*

Алгоритми для мультизірки та мультидерева

Повернемося до проблеми знаходження мінімального остового дерева з гілками (B-MST-проблеми). У відповідньому розділі дисертації розглядається розв'язний випадок цієї проблеми, а саме, випадок, коли граф G_ϕ є мультидерево. Розглянемо спочатку допоміжну проблему.

Нехай $H = (N, A)$ є орієнтований мультиграф. Деякі дуги в A згруповані в так звані *суміжні пари*: *Суміжна пара* включає дві дуги a_i і a_j, де $a_i = (v_s, v_t)$ і $a_j = (v_t, v_s)$ для де-якої пари вершин $v_s, v_t \in N$, $v_s \neq v_t$. Кожна дуга належить не більш, як до одної суміжної пари. Дуги, які не належать ні до одної суміжної пари, називаються *одинокими дугами*. Для $B \subseteq A$, нехай $Twin(B)$ означає множину всіх суміжних пар, що належать B, а $Sing(B)$ означає множину всіх одиноких дуг в B (ці дуги можуть мати суміжні дуги в A, але не в B), і нехай $Loop(B)$ означає множину всіх петель в B.

Для кожної дуги a в A, визначається вага $g(a)$, і для кожної суміжної пари $(a_i, a_j) \in Twin(A)$ – вага $g(a_i, a_j)$. *Вага Гайкова* (чи *g–вага*) мідмножини $B \subseteq A$ визначається як

$$g(B) = \sum_{(a_i,a_j)\in Twin(B)} g(a_i, a_j) + \sum_{a_k \in Sing(B)} g(a_k) + \sum_{a_l \in Loop(B)} g(a_l).$$

Говорять, що кожна дуга $(v_s, v_t) \in A$ *покриває* вершину v_t. Мідмножина $B \subseteq A$ є *точним покриттям* орієнтованого графа

H, якщо кожна вершина N покривається точно однією дугою із B.

Лема 24. *Нехай $H = (N, A)$ є орієнтований мультиграф, де кожна вершина має не менше однієї дуги, що входить в неї, і нехай g є вагова функція, на A $Twin(A)$. Тоді точне покриття B^* для H з мінімальною вагою Гайкова може бути знайдене за $O(|A|^2 \log |A|)$ елементарних операцій.*

Розглянемо тепер проблему пошуку мінімального остового дерева для мультизірки. Н. Гайков (О минимизации линейной формы на циклах, Рукопись предст. на деп. редкол. журнала *Изв. АН БССР*, 1980, ВИНИТИ 479-80 Деп.) запропонував звести проблему пошуку мінімального остового дерева з гілками до проблеми пошуку точного покриття з мінімальною g–вагою у відповідньому графі.

Теорема 25 *(Гайков). Нехай $G = (V, E)$ є Ейлерова мультизірка з $E = \{1, 2, \ldots, |E|\}$ така, що послідовність ребер $1, \ldots, |E|$ формує Ейлерів шлях. Нехай w є вагова функція на інтервалах $[i, \ldots, j]$, $1 \le i \le j \le |E|$, яка задовільняє умовам (6). Тоді остове дерево в G з мінімальною вагою гілок може бути знайдене за $O(|V|^2 + |E|)$ елементарних операцій.*

Гайков використав свій алгоритм для мультизірки і запропонував алгоритм складності $O(nm^5)$, (де m – кількість підциклів в оптимальному призначенні) для розв'язання ПК Монжа у випадку коли граф склейки є мультидеревом. В [24] для цього випадку запропоновано новий алгоритм складності $O(nm^2)$, який базується на наступній теоремі.

Теорема 26. *Нехай $G = (V, E)$ є Ейлерове мультидерево з $E = \{1, 2, \ldots, |E|\}$ таке, що послідовність ребер $1, \ldots, |E|$ формує Ейлерів шлях. Нехай w є вісова функція на інтервалах $[i, \ldots, j]$, $1 \le i \le j \le |E|$, яка задовільняє умовам (6). Тоді остове дерево з мінімальною вагою гілок може бути знайдене за $O(|V|^2|E|)$ елементарних операцій.*

4 Нижні оцінки, що базуються на розв'язних випадках ПК

4.1 Нижні оцінки, що базуються на матрицях Деміденка

У відповідньому розділі дисертації показано, як можна обчислити нижні межі, використовуючи симетричні матриці Деміденка (аналогічний підхід може бути використано і для несиметричних матриць).

Для довільної матриці C може бути побудована така матриця Деміденка $D \in \mathcal{D}$, що $C = D + A$, де A – невід'ємна матриця. Нехай ϕ_{opt} є оптимальний тур для матриці C, а τ_{opt} – оптимальний (пірамідальний) тур для матриці D. Звідси витікає, що

$$c(\phi_{opt}) = a(\phi_{opt}) + d(\phi_{opt}) \geq d(\phi_{opt}) \geq d(\tau_{opt}).$$

Це означає, що довжина мінімального туру для D є нижня оцінка для довжини оптимального туру в ПК з матрицею відстаней C. Матриця D будується зменшенням деяких елементів в C наступним чином.

Не існує умов Деміденка, які б включали тільки елементи $(n-1)$-го і n-го рядків (стовбців). Тому ці елементи залишаються незмінними. Елементи з $(n-2)$-го рядка пов'язані деякими умовами з елементами $(n-1)$-го рядка. Для того щоб вибрати елементи $a_{n-2,j} = c_{n-2,j} - d_{n-2,j}$ якнайменшими, обчислимо елементи $d_{n-2,j}$ для $j = 1, \ldots, n-3$ таким чином:

$$d_{n-2,j} := \min\{c_{n-2,j};\ d_{n-2,n} - d_{n-1,n} + d_{n-1,j}\}.$$

В $(n-3)$-му рядку, кожен елемент $d_{n-3,j}$ $(j = 1, \ldots, n-4)$ пов'язан умовами Деміденка з елементами $(n-2)$-го рядку. Тому, для $j = 1, \ldots, n-1$, $d_{n-3,j}$ можуть бути обчислені як

$$d_{n-3,j} := \min\{c_{n-3,j};\ \min\{d_{n-3,k} - d_{n-2,k} \mid k = n-1, n\} + d_{n-2,j}\}.$$

Продовжуючи таким чином, можна обчислити всі елементи в D.

4.2 Нижні оцінки, що базуються на матрицях Ван дер Феена

Симетрична матриця $C = (c_{ij})$ є *матрицею Ван дер Феена*, якщо $c_{ij} + c_{j+1,k} \le c_{ik} + c_{j,j+1}$ для всіх i, j, k таких, що $1 \le i < j < j+1 < k \le n$. Відомо, що ПК з матрицею Ван дер Феена є також пірамідально розв'язною(J.A.A. van der Veen, A new class of pyramidally solvable symmetric traveling salesman problems, *SIAM J. Disc. Math.* 7, 1994).

У відповідному розділі дисертації обгрунтовуються нижні межі, що можуть буту побудовані, використовуючи симетричні матриці Ван дер Феена. Як і в попередньому розділі, для довільної матриці C, може бути побудована матриця Ван дер Феена V така, що $C = V + A$, де A – невід'ємна матриця.

4.3 Реконструкція невідомих елементів в неповних матрицях Монжа

Хоча проблема розпізнавання неповних матриць Монжа і є NP–повною, загальний підхід, що дозволяє запобігти заборонені підматриці, в деяких випадках розпізнає такі матриці за поліноміальний час.

Нехай перестановки σ і τ, що розпізнають перевпорядковану неповну матрицю Монжа, побудовані. Чи можна реконструювати невідомі елементи таким чином, щоб отримати матрицю Монжа?

Із визначення матриці Монжа витікає, що для цього необхідно розв'язати систему із не більше як $4u$ лінійних нерівностей, де u – це кількість невідомих елементів. Нехай $d_{ij} := c_{\sigma(i)\tau(j)}$. Тоді для кожного невідомого елемента x_{ij} в (d_{ij}) у систему нерівностей необхідно включити наступні нерівності:

$$
\begin{aligned}
x_{ij} + d_{i+1,j+1} &\le d_{i,j+1} + d_{i+1,j} \\
x_{ij} + d_{i+1,j-1} &\ge d_{i,j-1} + d_{i+1,j} \\
x_{ij} + d_{i-1,j-1} &\le d_{i,j-1} + d_{i-1,j} \\
x_{ij} + d_{i-1,j+1} &\ge d_{i-1,j} + d_{i,j+1}
\end{aligned}
$$

Деякі елементи d_{kl} тут також можуть бути невідомими. Якщо розв'язок системи нерівностей існує, то далі треба розглядати ПК Монжа з новою матрицею. Можуть мати місце наступні випадки:

(*i*) Відомі алгоритми для ПК Монжа не спроможні знайти точне рішення за поліноміальний час.

(*ii*) Оптимальний тур знайдено, але він включає заборонені ребра.

(*iii*) Знайден оптимальний тур, який не має заборонених ребер.

Якщо має місце випадок (*iii*), то це означає, що знайдено точне рішення ПК з неповною матрицею Монжа. Якщо ж має місце випадок (*ii*), то довжина знайденого туру є нижня межа для довжини оптимального туру.

5 Експоненційні околи для ПК

У відповідному розділі дисертації розглядаються евристичні алгоритми для ПК, які базуються на розв'язних випадках, досліджених в дисертації. Ці евристики можуть бути охарактеризовані як алгоритми локального пошуку, що мають такі властивості: в околі з експоненційною кількістю турів оптимальний тур знаходиться за поліноміальний час.

5.1 Пірамідальні тури і експоненційні околи

Нехай $\Lambda_1(\varphi)$ є множина турів

$$\tau = < \varphi(i_1), \varphi(i_2), \ldots, \varphi(i_{k-1}), \varphi(n), \varphi(i_{k+1}), \ldots, \varphi(i_n) >$$

де $i_1 < i_2 < \ldots < i_{k-1} < n$ і $n > i_{k+1} > \ldots > i_n$. Будемо називати $\Lambda_1(\varphi)$ *пірамідальним околом* перестановки φ.

Зрозуміло, $\Lambda_1(\varphi)$ з $\varphi = \varepsilon$ є добре відома множина пірамідальних турів. Перестановку φ можна розглядати як

перестановку для перенумерації міст в ПК. Змінюючи φ, можна генерирувати різні околи з 2^{n-2} турами. Як було відмічено вище, оптимальний тур в такому околі може бути знайдено за $O(n^2)$ елементарних операцій.

5.2 Майже пірамідальні тури

У відповідньому розділі дисертації узагальнюються ідеї із попереднього розділу.

Нагадаємо процедуру для побудови пірамідального туру: починаємо з елементу 1. Елемент 2 може бути розташован перед або після елементу 1. Таким чином, якщо перші $k-1$ елементів вже розташовані, елемент k може бути розташований перед або після цих елементів.

Визначимо тепер *множину* Λ_2 майже пірамідальних турів таким чином. Знову, фіксуємо 1. Далі будуємо пірамідальний тур, але дозволяємо міняти місцями елементи $(2k, 2k+1)$ для $k = 1, 2, \ldots, \lfloor (n-1)/2 \rfloor$.

Наприклад, існує 6 можливостей для розташування 2, 3:

$$\langle \ldots 123 \ldots \rangle, \langle \ldots 132 \ldots \rangle, \langle \ldots 213 \ldots \rangle,$$
$$\langle \ldots 312 \ldots \rangle, \langle \ldots 231 \ldots \rangle, \langle \ldots 321 \ldots \rangle.$$

Твердження 27. *Оптимальний тур з Λ_2 може бути знайдено за $O(n^2)$ елементарних операцій.*

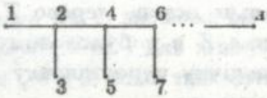
Окіл $\Lambda_2(\varphi)$ може також бути використано для побудови евристичних алгоритмів.

5.3 Оптимальні тури і добуток транспозицій

Нехай L є множина транспозицій

$$L = \{\langle i_1, j_1 \rangle, \langle i_2, j_2 \rangle, \ldots, \langle i_{n-1}, j_{n-1} \rangle\}$$

де $i_1, j_1, \ldots, i_{n-1}, j_{n-1}$ належать множині $I = \{1, 2, \ldots, n\}$.



Мал. 3: Граф Полья для $\Lambda_2(\varepsilon)$

Граф $T = (I, L)$ називається графом Полья для множини транспозицій L. Ми використовуємо тут визначення $\langle i, j\rangle$ і для транспозиції і для ребра в графі.

Твердження 28. *Нехай граф Полья T_1 є простим шляхом з множиною ребер*

$$L(T_1) = \{\langle 1,2\rangle, \langle 2,3\rangle, \ldots, \langle n-1, n\rangle\},$$

тоді множина всіх перестановок, що можуть бути побудовані перемноженням транспозицій $L(T_1)$, співпадає з множиною $\Lambda_1(\varepsilon)$.

Твердження 29. *Нехай граф Полья T_2 є граф, зображений на мал. 3 з множиною ребер*

$$L(T_2) = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 4,5\rangle, \langle 4,6\rangle, \ldots\},$$

тоді множина всіх перестановок, що можуть бути побудовані перемноженням транспозицій $L(T_2)$, співпадає з множиною $\Lambda_2(\varepsilon)$.

Нехай тепер граф Полья є довільним деревом T. Визначимо окіл $\Lambda_{tree}(T)$ як множину перестановок, які можуть бути отримані перемноженням транспозицій T. Відомо, що всі такі перестановки будуть циклічними, що робить цю множину цікавою для ПК. Більш того, виявляється, що множина $\Lambda_{tree}(T)$ має тісний зв'язок з відомим так званим алгоритмом дерева для ПК.

Алгоритм дерева для ПК може бути сформульовано таким чином: Знайти мінімальне остове дерево T, подвоїти ребра в T і знайти Ейлерів цикл Z в побудованому таким чином графі. Побудувати із Z циклічну перестановку (тур), відкидаючи ті вершини, що повторюються.

Відомо, що якщо відстані в матриці відстаней задовільняють умовам метрики, то рішення, отримане алгоритмом дерева, відрізняється від точного не більш ніж в два рази.

Наступна проблема виникає цілком природньо: На дане дерево T, знайти найкоротший тур серед всіх турів, що можуть бути побудовані алгоритмом дерева. Цю проблему будемо називати проблемою Оптимальної Реалізації Алгоритма Дерева. Хоча алгоритм дерева добре відомий, проблема оптимальної реалізації залишалась відкритою до цього часу. Більш того, Пападімітріоу і Вазірані (C.H. Papadimitriou and U.V. Vazirani, On two geometric problems related to the traveling salesman problem, *Journal of Algorithms* 5, 1984) зробили припущення, що ця проблема NP–повна.

Теорема 30. *Проблема Оптимальної Реалізації Алгоритма Дерева еквівалентна проблемі пошуку оптимального туру в околі* $\Lambda_{tree}(T)$.

6 Алгоритм дерева: узагальнення алгоритма динамічного програмування Хелда і Карпа для ПК

Алгоритм пошуку оптимального туру в околі $\Lambda_{tree}(T)$ описується у відповідній главі дисертації. Як відомо, дерева широко використовувались в алгоритмах для ПК. Оптимальна Реалізація Алгоритма Дерева (ОРАД) дає нам нові можливості як у використанні відомих алгоритмів, так і у побудові нових алгоритмів. Якщо дерево T є простим шляхом, то ОРАД є не що інше, як алгоритм пошуку оптимального пірамідального туру. Якщо дерево T є зіркою, то, як буде наголошено далі, ОРАД є не що інше, як відомий алгоритм динамічного програмування Хелда і Карпа для точного рішення ПК. Як бачимо, ОРАД комбінує два добре вивчених випадки ПК і, що може ще важливіше, відкриває простір для нових досліджень і для відшуку тих випадків, коли поліноміальна версія ОРАД може знаходити точні рішення.

6.1 Оптимальна реалізація алгоритма дерева

Теорема 31. *Оптимальний тур в $\Lambda_{tree}(T)$ може бути знайдено за $O(n^2 2^d + n^3)$ елементарних операцій, де d є максимальна ступінь вершин в дереві T. Для дерев спеціальної структури оптимальний тур може бути знайдений за $O(n^2 2^d)$ елементарних операцій.*

Висновок 32. *Для Евклідової ПК оптимальний тур серед всіх турів, що можуть бути побудовані алгоритмом дерева, може бути знайдений за поліноміальний час.*

Зауваження 1. Якщо ступені вершин в дереві обмежені константою, то ОРАД потребує тільки поліноміального часу. Це має місце, наприклад, коли розглядається Евклідова ПК (див. Висновок 32). Це означає, що згадане вище припущення Пападімітріоу і Вазірані про $NP-$повноту Евклідової ПК є хибне.

Зауваження 2. Узагальнення алгоритма Хелда і Карпа. Якщо дерево T є зіркою, то $\Lambda_{tree}(T)$ вміщує всі $(n-1)!$ циклові перестановки. Аналіз ОРАД показує, що в цьому випадку цей алгоритм є не що інше, як відомий алгоритм Хелда і Карпа для ПК (M.Held and R.M.Karp, A dynamic programming approach to sequencing problems, *SIAM Journal on Applied Mathematics* 10, 1962), який потребує $O(n^2 2^n)$ елементарних операцій.

Зауваження 3.

$$|\Lambda_{tree}(T)| = \prod_{i=1}^{n} deg(i)!, \text{ де } deg(i) \text{ є ступінь вершини } i \text{ в } T.$$

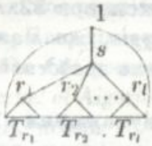
6.2 Ще дещо про ОРАД

Як було вказано вище, ОРАД комбінує два відомих алгоритми для ПК. У відповідньому розділі дисертації аналізуються інші властивості ОРАД.

ПК на графах Халіна. *Граф Халіна* будується таким чином: Погрузити довільне дерево T в площину і добавити нові ребра, які з'єднують в один цикл C всі висячі вершини T, таким чином, щоб новий граф $H = T \cup C$ залишився планарним.

Корнуелс та інші (G.Cornuejols, D.Naddef, W.R.Pulleyblank, Halin graphs and the traveling salesman problem, *Mathematical Programming 26*, 1983) довели, що ПК на графах Халіна може бути вирішена за лінійний час.

Твердження 33. *Оптимальна реалізація алгоритму дерева знаходить точне рішення ПК на графах Халіна.*



Алгоритм Крістофідеса і ОРАД. Крістофідес і, незалежно від нього, Сердюков запропонували евристичний алгоритм для ПК з метричною матрицею відстаней. Довжина туру, знайденого по алгоритму Крістофідеса, не перевищує довжину оптимального туру більш ніж в 1.5 рази. Ідея цього алгоритму близька до ідеї алгоритму дерева. До мінімального остового дерева добавляються ребра мінімального паросполучення і відшукується Ейлерів цикл у новому графі. Тур отримується із цього циклу відкиданням вершим, що повторюються. На відміну від алгоритма дерева, проблема пошуку оптимального туру серед всіх турів, що можуть бути побудовані алгоритмом Крістофідеса, є NP–повною, навіть для Евклідової ПК.

Але в деяких випадках ОРАД може знайти точне рішення серед всіх турів, що можуть бути побудовані алгоритмом Крістофідеса. На малюнку нижче зображене мінімальне остове дерево та два Ейлерівих графи, що були побудовані алгоритмом Крістофідеса. Граф (*ii*), на відміну від графа (*iii*), має чудові властивості: не існує двох циклів, які б мали спільне ребро. В цьому випадку такий граф може бути отримано з ейлерового циклу, який будується в алгоритмі дерева. Це означає, що ОРАД знайде в цьому випадку рішення не гірше, ніж оптимальний тур серед всіх турів, що можуть бути побудовані алгоритмом Крістофідеса.

(*i*) (*ii*) (*iii*)

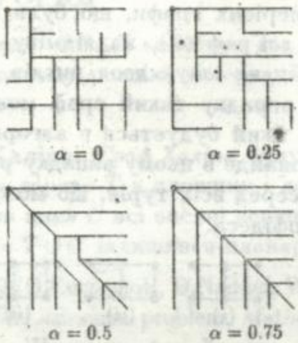
7 Евристичні алгоритми, що базуються на ОРАД

Зрозуміло, що мінімальне остове дерево не є єдиним можливим деревом, що може подаватися на вхід алгоритма дерева чи ОРАД; У відповідньому розділі дисертації пропонується підхід для генерирування дерев, які не дуже відрізняються від мінімального остового дерева. Це так звані *параметричні* або *α−дерева* і дерева для збільшених околів. Такий підхід дозволив побудувати нову евристику, яка спроможна змагатися з кращими евристичними алгоритмами для ПК.

7.1 Параметричні дерева

Розглянемо два алгоритми: алгоритм Пріма для мінімального остового дерева і алгоритм Дейкстри для дерева найкоротших шляхів. Основний крок в цих алгоритмах – це включення нової вершини в дерево. В обох алгоритмах для цього відшукується вершина з мінімальним ключем. В алгоритмі Пріма ключ $key_1(v)$ обчислюється як довжина мінімального ребра із вершини v до вершин дерева:

$$key_1(v) := min\{a_{vw}|w \in T\}.$$



Мал. 4: α–дерева

В алгоритмі Дейкстри ключ $key_2(v)$ обчислюється як найменша відстань від вершини v до стартової вершини s:

$$key_2(v) := min\{a_{vw} + dist(w) | w \in T\},$$

де $dist(w)$ - це довжина шляху в дереві від w до стартової вершини s.

Обидва алгоритми можуть бути поєднані, якщо обчислювати ключ $key(v)$ як

$$key(v) := min\{a[v, w] + \alpha \cdot dist(w) | w \in T\},$$

де $dist(w)$ - та ж сама довжина, що й в алгоритмі Дейкстри, а α - константа із відрізку $[0, 1]$. Зрозуміло, якщо $\alpha = 0$, то отримуємо алгоритм Пріма, а якщо $\alpha = 1$, то - алгоритм Дейкстри. Змінюючи α від 0 до 1, можна генерирувати остові дерева і використовувати їх в ОРАД для отримання турів комівояжера.

На мал. 4 зображені чотири дерева на решітці для різних α.

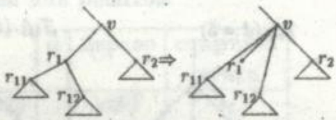
7.2 Збільшені околи

Нехай d є максимальна ступінь вершин в T. Зрозуміло, не всі вершини в T обов'язково мають таку ступінь. Спробуємо побудувати нове дерево T_1, в якому, як і раніше, ступені вершин не перевищували б d, але мало б місце таке включення $\Lambda_{tree}(T_1) \supseteq \Lambda_{tree}(T)$ і $|\Lambda_{tree}(T_1)| \geq |\Lambda_{tree}(T)|$.

Якщо таке дерево існує, будемо називати новий окіл *збільшеним околом*.

Лема 34. *Нехай вершина v в T має l синів $r_1, r_2, \ldots, r_l$. Нехай далі дерево T_1 побудовано із T за допомогою* операції збільшення ступеня*:*

вилучити всіх синів вершини r_{i_m} $(1 \leq m \leq l)$ і зробити їх синами вершини v.



Тоді має місце наступне включення:

$$\Lambda_{tree}(T_1) \supseteq \Lambda_{tree}(T).$$

Операція збільшення ступеню є коректною, якщо новый ступінь вершини v не перевищує d. Але, треба мати на увазі, що ми не можемо вилучати тільки частину синів вершини. Формально це зауваження може бути сформульовано таким чином:

Лема 35. *Нехай v в T має l синів $r_1, r_2, \ldots, r_l$. Нехай деякі (не* всі*) сини вершини r_m $(1 \leq m \leq l)$ були перепризначені як сини v чи другої вершини в T. Тоді для нового дерева T_1 має місце такий факт:*

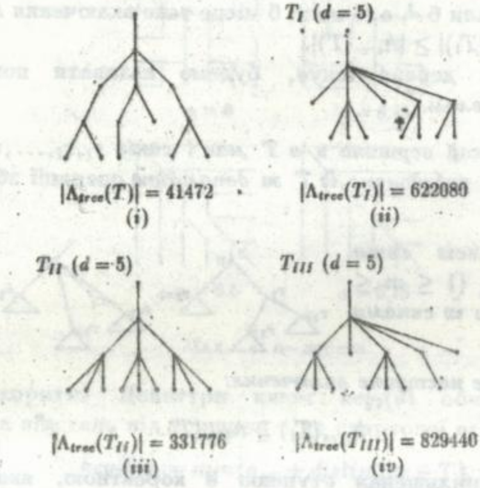
$$\Lambda_{tree}(T_1) \not\supseteq \Lambda_{tree}(T).$$

Із Леми 34 і Леми 35 негайно витікає наступна

Теорема 36. *Нехай T і T_1 є два кореневі дерева. Тоді*

$$\Lambda_{tree}(T_1) \supseteq \Lambda_{tree}(T),$$

якщо і тільки якщо T_1 може бути побудовано із T за допомогою операції збільшення ступеню.



Мал. 5: Дерева для збільшених околів

Неоднаразове використання операції збільшення ступеню може бути реалізовано по різному. Наприклад, це може бути зроблено простим 'жадним' алгоритмом під час перегляду дерева в прямому чи в зворотньому напрямку. На мал. 5 зображено дерево T і два дерева T_I і T_{II}, які були побудовані таким алгоритмом. На мал. 5 (iv) зображене дерево T_{III}, яке теж було побудоване з допомогою операції збільшення ступеню. Дерево T_{III} має окіл з найбільшою кількістю турів серед всіх дерев, що мають ступінь вершин не більше 5, і які можуть бути побудовані з T. Для фіксованого ступеню d таке дерево можна побудувати за поліноміальний час для довільного остового дерева T.

7.3 Комбінована евристика

Комбінуючи параметричні дерева і дерева для збільшених околів, ми побудували евристичний алгоритм для ПК. Змінюючи параметр α і будуючи дерева для збільшених околів при різних значеннях d, ми використовували ці дерева в ОРАД.

Для 17 прикладів, відомих з літератури для ПК, було знайдене таке значення параметру α, при якому ОРАД знаходить точне рішення.

Оптимальні параметри для решіток:

n	α	d	дерево	стартова точка
16	0.025		MST	s_1
36	0.025		MST	s_1
64	0.	5	T_{II}	s_1
100	0.	9	T_{II}	s_1

Оптимальні параметри для малих ПК:

Проблема	α	d	дерево	стартова точка
gr17	0.	4	T_I	s_1
gr21	0.	5	T_I	s_1
gr24	0.	7	T_I	s_1
dntz42	0.	5	T_I	s_1
gr48	0.075	9	T_I	s_2
hk48	0.05	7	T_I	s_2
th57	0.05	8	T_I	s_1^*
st70	0.2	5	T_{II}	s_2
s_1^* - стартова точка в дереві з іншим коренем				

Оптимальні параметри для ПК середнього розміру

Проблема	α	d	дерево	стартова точка
kroA100	0.075	9	T_I	s_1
kroB100	0.	6	T_{II}	s_1
kroC100	0.2	9	T_I	$s^* = 70$
kroD100	0.05	4	T_I	s_1
kroE100	0.	4	T_{II}	s_1

8 Висновки

В дисертації розглянуті розв'язні випадки класичної задачі комбінаторної оптимізації – проблеми комівояжера. Наголос зроблено на алгоритмах розпізнавання розв'язних випадків. Результати досліджень мають не тільки теоретичне значення: побудовані евристичні алгоритми ілюструють можливості нових алгоритмів, досліджених в роботі. Узагальнюючи основні результати дисертації, можна констатувати, що знайдено нові розв'язні випадки класичної ПК, сукупність яких можна кваліфікувати як нове вагоме досягнення у всебічному дослідженні класичної задачі комбінаторної оптимізації. Основні результати дисертації такі:

- Побудовано оптимальний алгоритм розпізнавання перевпорядкованих матриць Монжа. Цей алгоритм має важливе значення, оскільки матриці Монжа знайшли останнім часом широке використання в комбінаторній оптимізації.

- Для таких добре відомих розв'язних випадків ПК, як ПК з матрицями Кальмансона і Деміденка, знайдена геометрична інтерпретація. Як наслідок такого дослідження, побудовані алгоритми розпізнавання планарних матриць Кальмансона ($O(n^2)$ алгоритм) і планарних матриць Деміденка ($O(n^4)$ алгоритм).

- Знайдено новий розв'язний випадок Евклідової ПК – 'опукла-оболонка-і-пряма'. Алгоритм має найкращу із можливих оцінку кількості елементарних операцій – $O(n^2)$.

- Проблема тура Майстра була сформульована Пападімітріоу в 1993 році як добрий кандидат на $\Sigma_2 P$-повну проблему. В дисертації

  – доведено, що ПК з матрицею відстаней C має тур Майстра, якщо і тільки якщо C є матрицею Кальмансона;

  – побудовано $O(n^2 \log n)$ алгоритм для розпізнавання перевпорядкованих матриць Кальмансона.

- Як узагальнення техніки, розвиненої автором для розпізнавання перевпорядкованих матриць, запропоновано загальну схему, що дозволяє запобігти заборонені 2×2 підматриці. Один із прикладів використання цієї схеми – поліноміальний алгоритм розпізнавання перевпорядкованих матриць Супніка, що також запропонований в дисертації.

- Дана повна характеризація розв'язних випадків ПК Монжа. Ця проблема притягувала увагу вчених напротязі довгого часу. Вклад автора в дослідження цієї проблеми такий:

  – Розроблено алгоритм розпізнавання перевпорядкованих матриць Монжа (згадувалось вище).

  – Запропоновано алгоритм з лінійною пам'яттю для пошуку оптимального пірамідального туру.

  – Розроблено ефективні алгоритми для розв'язних випадків:

    * лінійний алгоритм для пошуку оптимального туру у випадку, коли граф склейки є мультишляхом, і, як наслідок – $O(n^2)$ алгоритм для випадку, коли граф склейки є мультициклом;

* $O(n^2 \log n)$ алгоритм для випадку, коли граф склейки є мультидеревом (відомий раніше алгоритм мав трудомісткість $O(n^6)$).

– Схема склейки Гілморе-Гоморі-Бурдюка-Трофімова узагальнена

* для нового класу матриць;
* для проблеми багатьох (не більше p) комівояжерів.

• Досліджено зв'язок між добутком транспозицій і множинами турів спеціальної структури. Як наслідок таких досліджень, отримано такі результати:

– Запропоновано новий алгоритм динамічного програмування для пошуку оптимального туру в експоненційному околі. Цей алгоритм поєднує два класичних алгоритми – алгоритм Хелда і Карпа і алгоритм пошуку оптимального пірамідального туру. Цей алгоритм має і інші чудові якості:

· *Алгоритм дерева* для ПК добре відомий. Але довго проблема пошуку найкоротшого туру серед всіх турів, що можуть бути побудовані алгоритмом дерева, залишалась відкритою. В дисертації доведено, що новий алгоритм динамічного програмування вирішує цю проблему за поліноміальний час у випадку, коли ступені вершин в дереві обмежені.

– Побудовано околи з експоненційною кількістю турів. Ці околи мають корисні властивості: оптимальний тур в такому околі може бути знайден за поліноміальний час. Більш того, для деяких розв'язних випадків ПК, досліджених в дисертації, оптимальні рішення можуть відшукуватись тільки в таких околах.

• Розв'язні випадки, досліджені в дисертації, використовуються для побудови нових евристичних алгоритмів.

Література

[1] Бурдюк В.Я., Дейнеко В.Г. Оптимальные круговые упорядочения (Задача о коммивояжере): Учеб. пособие по спецкурсу. Днепропетровск: ДГУ, 1983. – 102 с.

[2] Дейнеко В.Г., Филоненко В.Л. О восстановлении матриц специальной структуры // Актуальные проблемы ЭВМ и программирования: Сб. научн. тр. – Днепропетровск: ДГУ, 1979. – С. 43 – 45.

[3] Дейнеко В.Г. Об одной задаче комбинаторной оптимизации // Актуальные проблемы ЭВМ и программирования: Сб. научн. тр. – Днепропетровск: ДГУ, 1976. – С. 93 – 96.

[4] Дейнеко В.Г. Применение динамического программирования для решения специальных задач многих коммивояжеров // Исследование операций и АСУ: Сб. научн. тр. – Киев: Выща школа, 1978. – Вып. 14. – С. 47 – 50.

[5] Дейнеко В.Г. О восстановлении матриц с неизвестными элементами // Актуальные проблемы ЭВМ и программирования: Сб. научн. тр. – Днепропетровск: ДГУ, 1980. – С. 40 – 47.

[6] Дейнеко В.Г. Полиномиальный алгоритм нахождения оптимума в экспоненциальных окрестностях для задачи о коммивояжере. // Методы решения нелинейных задач и обработка данных: Сб. научн. тр. – Днепропетровск: ДГУ, 1985. – С. 40 – 42.

[7] Дейнеко В.Г. Рекурсивный алгоритм для задачи о коммивояжере // Методы решения нелинейных задач и обработка данных: Сб. научн. тр. – Днепропетровск: ДГУ, 1986. – С. 30 – 42.

[8] Дейнеко В.Г., Трофимов В.Н. О возможных значениях функционала в специальных задачах коммивояжера // Кибернетика. – 1981. №6. – С. 135 – 136.

[9] Дейнеко В.Г., Калька О.Ф. О программной реализации одного комбинаторного алгоритма // Актуальные проблемы ЭВМ и программирования: Сб. научн. тр. – Днепропетровск: ДГУ, 1981. – С. 36 – 38.

[10] Дейнеко В.Г. Специальный случай задачи о назначениях // Исследования по современным проблемам суммирования и приближения функций и их приложениям: Сб. научн. тр. – Днепропетровск: ДГУ, 1989. – С. 80 – 83.

[11] Дейнеко В.Г. Алгоритм генерирования случайных подстановок специальной структуры // Методы решения нелинейных задач и обработка данных: Сб. научн. тр. – Днепропетровск: ДГУ, 1983. – С. 30 – 37.

[12] Дейнеко В.Г. Некоторые методы и алгоритмы решения задач комбинаторной оптимизации: Автореф. дис.... канд. физ.-мат. наук. Минск: Институт Математики АН БССР, 1981. – 11 с.

[13] Бурдюк В.Я., Дейнеко В.Г. Экономный алгоритм решения специальных задач коммивояжера // Исследование операций и АСУ: Сб. научн. тр. – Киев: Выща школа, 1978. Вып. 12. – С. 77 – 81.

[14] Бурдюк В.Я., Дейнеко В.Г. О разрешимых случаях задачи многих коммивояжеров // Теория оптимальных решений: Сб. научн. тр. – Киев: Институт Кибернетики АН УССР, 1979. – С. 3 – 9.

[15] Борзова И.В., Дейнеко В.Г. К оптимизации функционалов на множестве остовных деревьев // Актуальные проблемы ЭВМ и программирования: Сб. научн. тр. – Днепропетровск: ДГУ, 1979. – С. 36 – 39.

[16] Данилович И.С., Дейнеко В.Г. Об оценке погрешности одного приближенного алгоритма // Методы решения нелинейных задач и обработка данных: Сб. научн. тр. – Днепропетровск: ДГУ, 1983. – С. 84 – 85.

[17] V.G. Deĭneko, R. Van Dal and G. Rote, The convex-hull-and-line traveling salesman problem: a solvable case, *Information Processing Letters* 51, 1994, 141–148.

[18] R.E. Burkard and V.G. Deĭneko, Polynomially solvable cases of the traveling salesman problem and a new exponential neighborhood, *Computing* 54, 1995, 191–211.

[19] V.G. Deĭneko, R. Rudolf and G.J. Woeginger, A general approach to avoiding 2×2 submatrices, *Computing* 52, 1994, 371–388.

[20] Дейнеко В.Г. Геометрические разрешимые случаи задачи о коммивояжере. Днепропетровск, 1988. – 21 с. – Деп. в ВИНИТИ, № 8630–В88.

[21] Дейнеко В.Г. Перемножение транспозиций, остовные деревья и задача о коммивояжере. Днепропетровск, 1987. – 29 с. – Деп. в ВИНИТИ, № 6577–В87.

[22] Дейнеко В.Г. Перемножение транспозиций и задача о коммивояжере; алгоритмы и вычислительный эксперимент. Днепропетровск, 1988. – 36 с. – Деп. в ВИНИТИ, № 3642–В88.

[23] Дейнеко В.Г. Выпуклая оболочка, три параллельные прямые и алгоритмы вставки для задачи о коммивояжере. Днепропетровск, 1989. – 31 с. – Деп. в ВИНИТИ, № 4766–В89.

[24] R.E. Burkard, V.G. Deĭneko and G.J. Woeginger, The traveling salesman problem on permuted Monge matrices, Technical Report 31, 1995, Institut fuer Mathematik, TU Graz, Austria.

[25] V.G. Deĭneko, R. Rudolf, J.A.A. van der Veen and G.J. Woeginger, Three Easy Special Cases of the Euclidean Travelling Salesman Problem, SFB Report 17, 1995, Institut fuer Mathematik, TU Graz, Austria.

[26] V.G. Deĭneko, R. Rudolf and G.J. Woeginger, On the Recognition of Permuted Supnick and Incomplete Monge Matrices, SFB Report 05, Spezialforschungsbereich "Optimierung und Kontrolle", TU Graz, Austria, to appear in *Acta Informatica*.

[27] V.G. Deĭneko, R. Rudolf and G.J. Woeginger, Sometimes Travelling is Easy: The Master Tour Problem, SFB Report 15, 1994, Institut fuer Mathematik, TU Graz, Austria.

[28] R.E. Burkard, V.G. Deĭneko, R. van Dal, J.A.A. van der Veen and G.J. Woeginger, Solvable cases of the traveling salesman problem: A survey of recent results, SFB Report 32, 1995, Institut fuer Mathematik, TU Graz, Austria.

[29] Дейнеко В.Г. Об оптимальной реализации алгоритма дерева решения задачи о коммивояжере // 8–я Всесоюзн. конф. по проблемам теорет. киберн.: Тезисы докладов. – Горький, 1988. – С. 105.

[30] Дейнеко В.Г. T–окрестности для алгоритма дерева решения задачи о коммивояжере // 9–я Всесоюзн. конф. по проблемам теорет. киберн.: Тезисы докладов. – Волгоград, 1991. – С. 37.

Deineko V.G. *Solvability in combinatorial optimization: Solvable cases of the traveling salesman problem and heuristics.* Thesis for obtaining a scientific degree "Doctor of physico-mathematical sciences" in the special field of 'Theoretical bases of informatics and cybernetics (mathematical cybernetics) – 01.05.01'. V.M.Glushkov Institute of Cybernetics, National Academy of Sciences, Kiev, Ukraine, 1995.

Solvable cases of the traveling salesman problem (TSP) are considered in the thesis, i.e. cases that the TSP can be solved in polynomial time. The emphasis is put on recognition algorithms for the well known solvable cases. Recognition algorithms for permuted Monge, Demidenko, Kalmanson, Supnick and incomplete Monge matrices are presented. Complete characterization of the TSP with a permuted Monge distance matrix is given. The well known subtour-patching strategy is generalized for a new class of matrices. A generalization of the classical dynamic programming algorithm for the TSP is presented. The new algorithm combines in a nice form two well known algorithms for the TSP: Held and Karp dynamic programming algorithm and algorithm for finding an optimal pyramidal tour. Moreover, the new algorithm is, in fact, an optimal realization of the minimal spanning tree heuristic for the TSP. Techniques developed for solving special cases of the TSP are used for constructing new heuristics.

Дейнеко В.Г. *"Разрешимость в комбинаторной оптимизации: Разрешимые случаи проблемы коммивояжера и эвристические алгоритмы"*. Диссертация на соискание учёной степени доктора физико-математических наук по специальности 01.05.01 – теоретические основы информатики и кибернетики (математическая кибернетика), Институт кибернетики им. В.М.Глушкова НАН Украины, Киев, 1995.

В работе рассматриваются разрешимые случаи задачи коммивояжера (ПК), т.е. случаи, когда задача может быть разрешена за полиномиальное время. Основное внимание уделено распознаванию разрешимых случаев. Предложены алгоритмы распознавания переупорядоченных матриц Монжа, Демиденко, Калмансона, Супника и неполных матриц Монжа. Дана полная характеризация ПК с переупорядоченными матрицами Монжа. Хорошо известная техника склеивания циклов обобщена на новый класс матриц. В работе представлено обобщение классического алгоритма динамического программирования для ПК. Новый алгоритм комбинирует два алгоритма для ПК: алгоритм динамического программирования Хелда и Карпа и алгоритм нахождения оптимального пирамидального тура. Более того, новый алгоритм является фактически оптимальной реализацией эвристического алгорита минимального связывающего дерева для ПК. Подходы, предложенные для решения специальных случаев, используются для построения эвристических алгоритмов.

Ключові слова: проблема комівояжера, поліноміальні алгоритми, розв'язний випадок, пірамідальний тур, алгоритм дерева, експоненційний окіл, евристичний алгоритм.

тир. ВТЧ зак. 1244-100.