Угорський алгоритм

Угорський алгоритм **-** алгоритм оптимізації, розв’язуючий задачу про призначення за поліноміальний час. Ідея цього методу розв’язання транспортної задачі вперше була запропонована угорським математиком Є. Егерварі 1931 року, тобто ще до розроблення загальної теорії лінійного програмування. Він був розроблений і опублікований ХаролдомКуном в 1955 році. Автор дав йому ім'я «угорський метод» у зв'язку з тим, що алгоритм в значній мірі заснований на більш ранніх роботах двох угорських математиків (Кеніга іЕгерварі).

Спочатку цей метод був розроблений для розв’язування специфічного виду транспортної задачі, а згодом узагальнений. Нині угорський метод є одним з найпоширеніших методів розв’язання транспортних задач.

Джеймс Манкрес в 1957 році помітив, що алгоритм є (строго) поліноміальним. Зцього часу алгоритм відомий також як алгоритм Куна - Манкреса або алгоритм Манкреса рішення задачі про призначення. Тимчасова складність оригінального алгоритму була*O(n4),* однак Едмондс і Карп (а також Томідзава незалежно від них)показали, що його можна модифікувати так, щоб досягти часу виконання *O(n3)*. Форд і Фалкерсон розповсюдили метод на загальні транспортні завдання. У 2006 році було виявлено, що К. Г. Якобі знайшов рішення задачі про призначення в *XIX* столітті і опублікував його в 1890 році на латині.

Специфічні особливості задачі про призначення дозволили розробити ефективний метод її вирішення, відомий як угорський метод.

Припустимо, є три працівника: Іван, Петро та Андрій. Потрібно розподілити між ними виконання трьох видів робіт (які ми назвемо A, B, C), кожен працівник повинен виконувати тільки один різновид робіт. Як це зробити так, щоб витратити найменшу суму грошей на оплату праці робітників? Спочатку необхідно побудувати матрицю вартостей робіт.

|  |  |  |  |
| --- | --- | --- | --- |
|  | А | В | С |
| Іван | 1000 грн. | 2000 грн. | 3000 грн. |
| Петро | 3000 грн. | 3000 грн. | 3000 грн. |
| Андрій | 3000 грн. | 3000 грн. | 2000 грн. |

Угорський алгоритм, застосований до наведеної вище таблиці дасть нам необхідний розподіл: Іван виконує роботу A, Петро - роботу B, Андрій - роботу С.

Алгоритм досить простий з погляду обчислень і може застосовуватися без упереджень навіть у разі виродженості плану.

Ідея методу полягає у здійсненні послідовного переходу від деякого недопустимого плану (не всі потреби задоволені і не вся продукція вивезена) до допустимого, що є розв’язком задачі. Цей перехід здійснюється за скінченну кількість ітерацій (але невідому до кінця обчислень), що пов’язані з перетвореннями матриці вартостей *C=(cij)* і поточного плану .

Назвемо умовно-оптимальним планом (псевдо планом) транспортної задачі таку сукупність невід’ємних чисел , яка задовольняє задану систему нерівностей:





і такі наступні умови для змінних двоїстої задачі — потенціалів:

*xij=0*, якщо *ui+vj-cij<0.*

Мірою недопустимості (умовою неоптимальності) умовно-оптимального плану може бути наявність різниці між сумою всіх запасів (чи потреб, що те саме) і сумою всіх перевезень умовно-оптимального плану, тобто:



Зрозуміло, що чим менша нев’язка*Δ,* тим ближче умовно-оптимальний план до найкращого плану транспортної задачі, а у разі, коли *Δ* = 0, він збігається з оптимальним планом.

Звідси легко збагнути ідею розглядуваного методу розв’я­зування транспортної задачі: починаючи з деякого початкового плану задачі, подвійної до транспортної , можна знайти послідовність оптимальних планів ряду допоміжних задач на мінімізацію за обмежень і , кожний наступний план якої надає нев’язціменшого значення у зіставленні з попереднім, а останній план цієї послідовності надає нев’язці нульового значення, збігаючись у такий спосіб з оптимальним планом транспортної задачі.

Отже, кожна ітерація методу означатиме розв’язування допоміжної задачі — і зменшення при цьому значення цільової функції  порівняно з попереднім розв’язком цієї задачі.

Щоб сформулювати допоміжну задачу, треба, крім використання величин  і , що їх містить задана транспортна задача, побудувати ще деякий план двоїстої задачі , . Для початку першої ітерації це легко зробити, узявши, наприклад:

, 

причому даний план задовольняє умову:

+.

а також у кожному рядку матриці перевезень унаслідок такого вибору потенціалів виконуватиметься хоча б одна рівність виду +. Справді, взявши для *i0*-го рядка в правій частині

+,

, дістанемо:

.

У наступних ітераціях утворену систему потенціалів змінюємо, але так, що вона завжди залишається планом подвійної задачі.

Наведені вище обмеження для змінних двоїстої задачі:

, якщо ;

, якщо 

означають, що клітини, в яких для визначеної на k-му кроці системи потенціалів  виконується строга нерівність , не заповнюють. Отже, розв’язуючи задачу, будемо використовувати лише ті клітини, для яких .

Зауважимо, що мінімізація цільової функції () рівнозначна максимізації другого її доданка



при тій самій системі обмежень. Зрозуміло, що , а при  матимемо: .

Виходячи з наведених теоретичних засад, розглянемо алгоритм угорського методу:

1. Побудова допоміжної задачі з цільовою функцією та умовами , 

2. Побудова початкового опорного плану допоміжної задачі, що отримана на попередньому кроці алгоритму, одним з відомих методів.

3. Відшукання оптимального плану допоміжної задачі.

3.1. Збільшення значення . Визначають рядки, де сума перевезень по рядку менша від запасів, а за допомогою них — стовпці, які мають у вибраному рядку не заборонені для перевезень клітини. Вибрані рядки і стовпці позначають так:

, ,

 та .

Потім , .

3.2. Визначення клітин, значення перевезень в яких *xij* необхідно змінити. Послідовність цих клітин повинна утворювати деякий ланцюг, елементи якого є в позначених рядках та колонках і за яким можна перенести лишок запасу деякого -го рядка, що був позначений першим, у *js*-ту колонку, позначену останньою.

У загальному випадку послідовність має вигляд:

.

У знайденому ланцюзі позначаємо першу його клітину з кінця знаком «плюс», а інші за чергою знаками «плюс» і «мінус». Знайдемо величину:

.

Як видно з алгоритму,  дорівнює меншій з двох величин: найменшого елемента ланцюга, що позначений знаком «мінус», і невикористаного запасу в позначеному першим пунктом відправленні. Величина  є незадоволеною потребою в позначеному наприкінці пункті доставки. Зсуву по ланцюгу підлягає мен­ша з цих величин, що й приводить до наведеної формули розрахунку *θ*.

4. Перехід до наступної допоміжної задачі, оптимальний план якої буде ближчим до оптимального плану початкової транспорт­ної задачі.

У кінцевій таблиці розв’язаної допоміжної задачі позначені колонки мають баланс суми перевезень по колонці і потреб у відповідних пунктах. Можна легко помітити, що заборону на перевезення слід знімати з тих клітин, які не належать до позначених колонок. Водночас у тій самій кінцевій таблиці попередньої допоміжної задачі з-поміж позначених є рядки, в яких немає балансу суми перевезень і запасів, так що шукана клітина міститиметься серед позначених рядків в остаточній таблиці.

Позначимо множину позначених рядків через , а множину позначених колонок — через J (у кінцевій таблиці, що містить розв’язок допоміжної задачі). Знайдемо величину:

,

тобто найменше значення різниці, що стоїть у дужках серед заборонених клітин, які містяться в позначених рядках і непозначених колонках;  строго більше від нуля, оскільки в іншому разі відповідна клітина не була б забороненою для перевезень, і її колонку можна було б позначити за допомогою того позначеного рядка, якому належить ця клітина, що суперечить умові . Зрозуміло, що, збільшивши, наприклад, потенціал , який входить у формулу (5.37), на величину , тим самим перетворюють відповідну клітину (в якій ) у вільну для перевезень. Звичайно, таких клітин, що задовольняють умову (5.26), може бути більше, ніж одна. Тому в усіх позначених рядках збільшуємо потенціали  на , а щоб зберегти попередню множину клітин вільних для перевезень, величину  віднімаємо від потенціалів позначених колонок . Решту потенціалів залишаємо незмінною. Тут слід підкреслити, що згідно з алгоритмом розв’язання допоміжної задачі всі вільні для перевезень клітини, які є в позначених рядках, обов’язково містяться і в позначених колонках. Отже, нову систему потенціалів знаходимо, змінюючи попередню за формулами:





Нова система потенціалів визначає нову множину клітин, заборонених для перевезень .

, .

Визначивши , знову формулюємо і розв’язуємо допоміжну задачу.

5. Повторення кроків 2—4 до відшукання оптимального плану початкової транспортної задачі.

Правила розв’язування задачу горським методом

Сформулюємо правила розв’язування допоміжної задачі, одночасно демонструючи їх на відповідному прикладі, і даючи в разі потреби відповідні пояснення.

Приклад розв’язання угорським методом транспортну задачу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Ai* | *Bj* | | | |
| *b*1 = 50 | *b*2 = 130 | *b*3 = 80 | *b*4 = 40 |
| *a*1 = 110 | 7 | 2 | 4 | 2 |
| *a*2 = 40 | 1 | 2 | 4 | 1 |
| *a*3 = 80 | 5 | 3 | 2 | 4 |
| *a*4 = 70 | 7 | 3 | 3 | 9 |

Розв’язання. Визначаємо початкову систему потенціалів для побудови першої допоміжної задачі, використовуючи (5.23):

, .

Маємо:

,

,

,

.

,

,

,

.

Розраховуємо величини  і записуємо їх у вигляді таблиці:

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 0 | 2 | 0 |
| 0 | 1 | 3 | 0 |
| 3 | 1 | 0 | 2 |
| 4 | 0 | 0 | 6 |

Клітини побудованої таблиці з елементами (різницями), відмінними від нуля, будуть забороненими для перевезень, чим гарантується виконання умов стосовно потенціалів, а клітини з нулями можна використати для знаходження плану допоміжної задачі.

Побудуємо початковий план допоміжної задачі, наприклад, методом північно-західного кута, враховуючи обмеження, накладені забороною зазначених перевезень (відповідні клітини виділені сірим кольором).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Ai* | *Bj* | | | | *αі* | λі |
| *b*1 = 50 | *b*2 = 130 | *b*3 = 80 | *b*4 = 40 |
| *a*1 = 110 |  | 2  110 |  | 2  0 |  |  |
| *a*2 = 40 | 1  40 |  |  | 1  0 |  |  |
| *a*3 = 80 |  |  | 2  80 |  |  |  |
| *a*4 = 70 |  | 3  20 | 3  0 |  | 50 | 0 |
| *βj* |  | 50 | 50 |  |  |  |
| *μj* |  | 4 | 4 |  |  |  |

Загальний обсяг перевезень за початковим планом  менший на 50 одиниць від обсягу всіх запасів (потреб): .

Спробуємо змінити визначений план для збільшення значення . Це можна зробити лише за рахунок тих рядків, де запас залишився невикористаним. Отже, позначимо ті рядки, де сума перевезень по рядку згідно з початковим планом менша від відповідного запасу: . Позначаємо їх двома числами. Одне з них відповідає значенню  і позначається символом , а друге є нулем і позначається символом , де і — номер вибраного рядка. У нашому прикладі таким буде лише один — четвертий рядок. Отже,  і, оскільки нуль відповідатиме четвертому рядку, то маємо: .

Взагалі таких рядків може бути кілька. За допомогою кожного позначеного рядка (в нашому прикладі четвертого) позначаємо колонки, які мають у позначеному рядку не заборонені для перевезень клітини, двома символами (5.31):  та .

У четвертому рядку не заборонені для перевезень клітини належать другій та третій колонкам, які позначаємо відповідними символами: , ; та , . Позначені колонки (друга і третя) використовуються для позначення ще не позначених рядків, які мають у даній позначеній колонці клітини з відмінними від нуля перевезеннями (*xij>0*).

Позначення відповідають умові (): , .

У такий спосіб далі позначаємо перший рядок числами:

, ,

а третій , .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Ai* | *Bj* | | | | *αі* | *λі* |
| *b1* = 50 | *b2* = 130 | *b3* = 80 | *b4* = 40 |
| *a1* = 110  +  +  – |  | 2  110 |  | 2  0 | 50 | 2 |
| *a2* = 40 | 1  40 |  |  | 1  0 |  |  |
| *a3* = 80 |  |  | 2  80 |  | 50 | 3 |
| *a4* = 70 |  | 3  20 | 3  0 |  | 50 | 0 |
| *βj* |  | 50 | 50 | 50 |  |  |
| *μj* |  | 4 | 4 | 1 |  |  |

Знову позначені рядки використовуємо для позначення за фор­мулами () нових, ще не позначених колонок, поки або не знайдеться колонка, сума перевезень якої менша від потреби відповідного пункту доставки, або процес позначень не можна буде продовжити. Другий випадок означає, що план змінити не можна, тобто він оптимальний. У першому разі продовжуємо алгоритм.

У нашому прикладі знайдено колонку, сума перевезень якої менша від потреби відповідного пункту. Це четверта колонка, яку і позначаємо числами:

, .

Визначаємо послідовність клітин, значення перевезень в яких слід змінити. Ця послідовність має утворювати деякий ланцюг, елементи якого мусять бути в позначених рядках та колонках, і за яким можна перенести лишок запасу деякого -го рядка, що був позначений першим, у -ту колонку, позначену останньою.

У нашому прикладі такою послідовністю клітин є: *А*4*В*2, *А*1*В*2, *А*1*В*4,

а .

Далі використовуємо формулу:



Отже, до перевезень *x42* і *x14* додаємо *θ* = 40, а від перевезення *x12* віднімаємо цю саму величину. Матимемо таблицю:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Ai* | *Bj* | | | | *αі* | *λі* |
| *b1* = 50 | *b2* = 130 | *b3* = 80 | *b4* = 40 |
| *a1* = 110 |  | 2  70 |  | 2  40 | 10 | 2 |
| *a2* = 40 | 1  40 |  |  | 1  0 | 40 |  |
| *a3* = 80 |  |  | 2  80 |  | 10 | 3 |
| *a4* = 70 |  | 3  60 | 3  0 |  | 10 | 0 |
| βj |  | 10 | 10 | 10 |  |  |
| μj |  | 4 | 4 | 1 |  |  |

Повторюючи процес позначень, починаючи з четвертого рядка, легко помітити, що він закінчується знову на четвертій колон­ці, де досягнута рівність суми перевезень колонки потребам. Отже, план допоміжної задачі оптимальний.

Зауважимо, що оптимальне значення лінійної функції  задачі  на 40 одиниць більше від початкового; збільшення лінійної функції забезпечується ациклічністю ланцюга і більшою на одиницю кількістю додатних (позначених знаком «плюс») клітин у ньому.

Перейдемо до нової допоміжної задачі, оптимальний план якої буде ближчим до оптимального плану заданої транспортної задачі; цільова функція допоміжної задачі , яка визначає загальний обсяг перевезених вантажів, при цьому збільшується. Для нашого прикладу цілком зрозуміло, що рівності  (тобто повного перевезення вантажу) можна було б досягти, якби, наприклад, зняти з *А*4*В*1 або з якоїсь іншої клітини першої колонки заборону на перевезення і здійснити перевезення  (*і* = 1, 3, 4). Оскільки згадана заборона накладена попередньою системою потенціалів , то зняти її можна, лише замінивши поперед­ню систему потенціалів новою, але такою, щоб зберігалася сукупність незаборонених для перевезень клітин, знайдена на попередньому етапі.

Для наведеного прикладу множина клітин (*i, j*), для яких , , така: *А*1*В*1, *А*3*В*1, *А*4*В*1, причому  досягається в клітині *А*3*В*1.

Знайдемо нову систему потенціалів:

,

,

,

,

,

,

,

.

Отже, тепер забороненими для перевезень клітинами будуть ті, де , що запишемо в таблицю:

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 | 2 | 0 |
| 0 | 4 | 6 | 3 |
| 0 | 1 | 0 | 2 |
| 1 | 0 | 0 | 6 |

Визначаємо початковий план і розв’язуємо нову допоміжну задачу:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Ai* | *Bj* | | | | *αі* | *λі* |
| *b*1 = 50 | *b*2 = 130 | *b*3 = 80 | *b4*= 40 |
| *a*1 = 110  +  +  – |  | 2  110 |  | 2  0 | 40 | 2 |
| *a*2 = 40 | 1  40 |  |  |  |  |  |
| *a*3 = 80 | 5  10 |  | 2  70 |  | 40 | 3 |
| *a*4 = 70 |  | 3  20 | 3  10 |  | 40 | 0 |
| *βj* |  | 40 | 40 |  |  |  |
| *μj* |  | 4 | 4 |  |  |  |

Зробивши зсув на 40 одиниць  по ланцюгу *А4В2, А1В2, А1В4,* матимемо такий план:

.

Цей план оптимальний, оскільки , тобто .

Зауважимо, що даний алгоритм застосовується і для розв’язування відкритих транспортних задач без введення фіктив­них пунктів. Для цього досить задовольнити ту групу обмежень, яка має виконуватись у вигляді строгих рівностей, і так підібрати систему потенціалів на останньому кроці, щоб тим обмеженням, які за знайденого плану є строгими нерівностями, відповідали нульові потенціали.

Задача про призначення

ЗАДАЧА ПРО ПРИЗНАЧЕННЯ - вид задачі лінійного програмування, за допомогою якої вирішуються питання типу: як розподілити робітників з верстатів, щоб загальна вироблення було найбільшим і затрати на заробітну плату найменшими (оскільки для кожної комбінації "робітник - верстат" характерна своя продуктивність праці ), як найкращим чином розподілити екіпажі літаків, як призначити людей на різні посади (звідси і назва завдання) і т. д.

Математично такі завдання - окремий випадок розподільних задач з тією особливістю, що в них обсяги готівки та потрібних для виконання кожної роботи ресурсів рівні одиниці, тобто *aj = bj= 1*, і усі *xij= 1*, якщо працівник *i* призначено на роботу *j* , або нулю в інших випадках. Інакше кажучи, для виконання кожної роботи витрачається тільки один вид ресурсу, а кожен ресурс може бути використаний на одній роботі: ресурси неподільні між роботами, а роботи - між ресурсами. Вихідні дані групуються в таблиці, яка називається матрицею оцінок, результати - в матриці призначень.

Кількість можливих варіантів призначень рівні факторіалу числа робіт і ресурсів і величезне навіть у невеликій задачи.. Тому для знаходження оптимального варіанту застосовують спеціальні алгоритми. Серед них особливо ефективний угорський метод.

Алгоритм:

1. Виконо попереднє перетворення С —» Б.

2. Вибір «О».

2.1. Відмітити «\*» будь-яку «0» «0» в 1-му стовпці матриці Б.

Відмітити «\*» будь-яку «0» » у 2-му стовпці матриці Б, але не лежачий в тому рядку, що і «0 першого стовпця.

Відмітити «\*» будь-який «*О*» в 3-му стовпці матриці Б, але не лежачий в тих же рядках, що і «0» першого і другого стовпців і т. д.

2.2. Якщо «*\**» відмічене п «*О*», то задача розв'язана. Відповідні вибраним елементам матриці Б {*йу* = 0\*) еквівалентні хi = 1. Призначення об'єктів на роботу виконано. Розв'язання задачі закінчити.

2.3. Якщо «*О\**» менше ніж п, то перейти до п. 3.

3. Вибір нових «*О*».

3.1. Відмітимо знаком «+» стовпці, в яких міститься «О\*» і назвемо ці стовпці вибраними (пізніше з'являться вибрані рядки, коли будуть вибиратися нулі-претенденти — «*О'*»). Елементи, що стоять на перетині невибраних стовпців і невибраних рядків, називаються невибраними елементами. Інші елементи вибрані.

3.2. Якщо в матриці немає невибраних «*О*», то знайти мінімальний елемент серед невибраних елементів матриці і відняти його з елементів невибраних рядків, а потім додати до елементів вибраних стовпців (т. ч. збережуться значення*О\**). Отримаємо еквівалентну матрицю, що містить принаймні один невибраний нуль. Помітки не знімати. Здійснюється, т. ч., мінімізація рішення по мінімально можливих приростах *Б(х).* Перейти до п. 3.3.

3.3. Якщо в матриці є невибрані нулі, то, переглядаючи невибрані стовпці зліва направо, виберемо перший нуль з непомічених «\*»-й або «'» і помітимо його «'». Цей нуль — претендент. Помітити рядок «+» (праворуч) — рядок вибраний. Якщо в рядку, де вибраний «*О'*» немає «*0\*»,* то перейти до п. 5 (вдалося вибрати на один «*0*» більше). Якщо ж у цьому рядку є «*0\**», то перейти до п. 4 (не вдалося вибрати більше нулів, так як «*О'*» замшить «*0\**»).

4. Звільнити (зняти «*+*») стовпець, що містить «*0\**», той, що лежить в одному рядку з «*О'*». Перейти до п. 3.2.

5. (У рядку, де знаходиться претендент *О'*, немає *0\**). Починаючи з *0'* останнього претендента будуємо ланцюжок з нулів: від «*О'*» до «*0\**», у тому ж стовпці, де і «*О'*», від «*0\**» по рядку до наступному «*О'*», від цього «*О'*» до наступного «*0\**» по стовпцю і т. д. поки ланцюжок не обірветься на деякому 0' (можливо, вона обірветься на першому ж *О'*). У ланцюжку від число «*О'*» більше числа «*0\**», тобто на один 0 вибрано більше. Зняти «*\**» у *0\**, що входять в ланцюжок, а «*'*» замінити на «\*». Зняти всі помітки і перейти до п. 2.2.

Ланцюжок тут будується для того, щоб стежити за «правильним» розташуванням вибираних нулів (по 1 в рядку і стовпці).

висновки

Спрощений метод розв'язання задачі про призначення та угорський метод розв'язання задачі про призначення дає можливість розподілу деякого числа робіт між таким же числом виконавців за умови взаємно-однозначної відповідності між безліччю робіт і виконавців.

Значення алгоритму, що розробляється, полягає в тому, щоб послідовно переходячи від одного вибору нулів до іншого, що містить на один нуль більше ніж попередній, побудувати повний вибір (план розподілу).

Якщо, перебравши всі нулі матриці, ми не доб'ємося успіху, то потрібно перетворити матрицю так, щоб в ній з'явилося більше нулів (потенційних претендентів на включення в список вибраних). При цьому на кожному кроці необхідно перейти до вибору з великим числом нулів.

Математична постановка задачі про оптимальні призначення

Є *n*видів робіт та n кандидатів для їх виконання (виконавців).

Вважається, що кожен з кандидатів *i=1,…,n*може виконувати будь-яку роботу *j=1,…n*, при цьому*cij* — витрати, пов'язані з призначенням*i* - го кандидата на j - й вид роботи. Необхiдно так розподілити кандидатів на виконання робіт, щоб кожен з кандидатів одержав єдине призначення, кожна з робіт одержалаєдиного виконавця i сумарні витрати, пов'язані з призначеннями, були мінімальними.

Це типова комбінаторна задача. Її розв'язком є деяке переставлення

Чисел *1,….n*. Число переставлень дорівнює *n!,* тому при великих n розв'язати цю задачу шляхом прямого перебирання усіх можливих переставлень неможливо. Однак сформульовану задачу можна записати у вигляді задачі лінійного програмування (ЗЛП) з цiлочисельними змінними. Дiйсно, нехай , якщо *i* - й виконавець призначається на *j* - ту роботу, інакше *xij=0*. Тоді математична модель задачі про оптимальні призначення приймає таку форму:

знайти матрицю *X = ||xij||,I,j=1,…,n,* що мінімізує цільову функцію

при виконанні обмежень

*xij=0 або 1, i,j=1,…,n.*

де *cij*— витрати, пов'язані з використанням *i* -го виконавця для виконання *j-ї* роботи. Елементи *cij* утворюють матрицю витрат *C*.

У відповідності з постановкою цієї задачі, розв'язати її — значить, іншими словами, вибрати у матриці *Cn* елементів, по одному з кожного рядка (рядки відповідають виконавцям) і кожного стовпця (стовпці відповідають роботам) так, щоб сума вибраних елементів, яка дорівнює сумарним витратам, пов'язаним з призначеннями, була найменшою порівняно з її значеннями при всіх інших таких призначеннях.

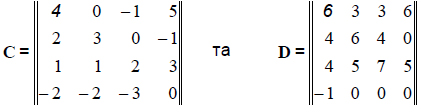
Зауважимо, що задачі математичного програмування, в яких на змінні накладаються умови *xij=0 або 1, i,j=1,…,n,* називаються задачами з булевими змінними. Отже, задача про оптимальні призначення є задачею лінійного програмування (ЗЛП) з булевими змінними. Крім того, цю задачу можна розглядати як частинний випадок транспортної задачі лінійного програмування. Дійсно, якщо умову *xij=0 або 1, I,j=1,…,n,* замінити умовою невiд'ємностi змінних, то*─ xij=0 або 1, i,j=1,…,n,* перетворюється у звичайну транспортну задачу, в якої об'єми виробництва та об'єми споживання є цілочисельними і рівними одиниці. Якщо розв'язати цю задачу методом потенціалів, або іншим методом, що забезпечує цiлочисельний оптимальний розв'язок при цiлочисельних об'ємах виробництва та споживання, то одержаний розв'язок буде автоматично задовольняти не враховане обмеження *xij=0 або 1, i,j=1,…,n.*

Проте специфіка цієї задачі дозволяє розв'язати її більш простими методами, ніж метод потенціалів. Такими методами є, наприклад, угорський метод.

Квадратні матриці *C=||cij||* та *D=||dij|| (i,j=1,…,n)*будемо називати еквівалентними, якщо елементи однієї з них одержуються з елементів другої шляхом додавання певних чисел до кожного рядка i кожного стовпця. Ці числа можуть бути різними для різних рядків та стовпцiв. Отже, матриці *C*та*D* еквівалентні, якщо, наприклад,

Dij=cij+ai+βj, i,j=1,…,n.

Зрозуміло, що ця властивість є взаємною.



є еквівалентними, бо матриця D одержується з матриці C додаванням до рядків чисел 0, 0, 1, –1 i додаванням до стовпців чисел 2, 3, 4, 1, відповідно.

**Теорема 4.** Оптимальні розв'язки задач про оптимальні призначення з еквівалентними матрицями витрат співпадають.

Доведення.Нехай матриця *C* еквівалентна матриці *D* і *(1,j1), (2,j2),…,(n,jn)* — оптимальні призначення у задачі з матрицею витрат C, тобто *i*-й виконавець призначається на виконання роботи *j1,i=1,…,n*. Припустимо, що це призначення не є оптимальним у задачі з матрицею витрат *D*.

Нехай *(1,j`1), (2,j`2),…,(n,j`n)*— оптимальні призначення у задачі з матрицею витрат *D*, причому

Звідси маємо

або

Проте

бо в обох випадках це є сума чисел, які додаються до стовпцiв матриці С. Отже маємо, що

Що протирічить тому, що для задачі з матрицею витрат *С* призначення (*i,ji*)*,i=1,…,n,* є оптимальним.

Оскільки властивість еквівалентності матриць є взаємною, то оптимальні призначення у задачі з матрицею витрат *C* співпадають з оптимальними призначеннями у задачі з матрицею витрат *D*. Доведення завершене.

Доведена теорема дозволяє, якщо це необхідно, переходити від даної задачі про оптимальні призначення до задачі з еквівалентною матрицею витрат. Тому вихідну задачу завжди можна звести до задачі про оптимальні призначення з матрицею витрат, яка має лише невід'ємні елементи. Оскільки найменше можливе значення суми *n* елементів такої матриці, очевидно, дорівнює нулю, то задача зводиться до вибору у матриці витрат, або еквівалентній їй, *n*нульових елементів, по одному в кожному рядку i кожному стовпці. В цьому, власне, полягає неформальний зміст алгоритму угорського методу.

Висновок

Вперше ідея угорського методу для розв’язання транспортної задачи була запропонована 1931 року, угорським математиком Є. Егерварі . Спочатку цей метод був розроблений для розв’язування специфічного виду транспортної задачі, а згодом узагальнений. Нині угорський метод є одним з найпоширеніших методів розв’язання транспортних задач.

Алгоритм виявився досить простим з погляду обчислень і може застосовуватися без упереджень навіть у разі виродженості плану.

Угорський метод найбільш ефективний при вирішенні транспортних задач з цілочисельними обсягами виробництва і споживання. У цьому випадку число ітерацій не перевищує величини *D0 / 2* (*D0* - сумарна невязка підготовчого етапу).

Перевагою угорського методу є можливість оцінювати близькість результату кожної з ітерацій до оптимального плану перевезень. Це дозволяє контролювати процес обчислень і припинити його при досягненні певних точностних показників. Дана властивість вигідна для задач великої розмірності.

Угорський метод розв'язання задачі про призначення дає можливість розподілу деякого числа робіт між таким же числом виконавців за умови взаємно - однозначної відповідності між безліччю робіт і виконавців.

Значення алгоритму, що розробляється, полягає в тому, щоб послідовно переходячи від одного вибору нулів до іншого, що містить на один нуль більше ніж попередній, побудувати повний вибір (план розподілу).

Якщо, перебравши всі нулі матриці, ми не доб'ємося успіху, то потрібно перетворити матрицю так, щоб в ній з'явилося більше нулів (потенційних претендентів на включення в список вибраних). При цьому на кожному кроці необхідно перейти до вибору з великим числом нулів.

Список літератури

1. Системы автоматизированного проектирования. В 9-ти кн.Учебное пособие для вузов. Под редакцией Норенкова И.П. М.: Высш. шк., 1986.
2. Норенков И.П. Введение в [автоматизированное проектирование](http://ua-referat.com/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) технических устройств и систем. Учебное пособие для вузов. - М.: Высш. шк., 1986.
3. П. Шеннен и др. [Математика](http://ua-referat.com/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) и [САПР](http://ua-referat.com/%D0%A1%D0%90%D0%9F%D0%A0). т.1. М.: Мир, 1988.
4. Батищев Д.И. Методы оптимального проектирования. М.: Радио и связь, 1984.
5. Системы автоматизированного проектирования в радиоэлектронике. Справочник. М.: Радио и связь, 1986.
6. Погребной В.К. О декомпозиции графов на классы изоморфных подграфов. В кн.: Вопросы программирования и автоматизации проектирования. Изд. ТГУ, 1979, с. 82-96.
7. Петренко А.И. Основы автоматизации проектирования. К.: Техника, 1982. - 295 с.
8. Ильин В.Н.. Основы автоматизации схемотехнического проектирования. Г.: Энергия, 1979. - 392 с.
9. Демидович Б.П., Марон И.А. Основы вычислительной математики. Г.: Изд-во «Наука», 1966. - 664 с.
10. Разевиг В.Д. Система сквозного проектирования электронных устройств DesignLab 8.0.- М.: Изд-во «Солон»,1999. - 698 с.
11. Ермольев Ю.М., Ляшко И.И., Михалевич В.С., Тюптя В.И. Математические методы исследования операций. К.: Вища школа, 1979.
12. ЗайченкоЮ.П., ШумиловаС.А. Исследование операций. Сборник задач. К.: Вища школа, 1990.
13. Корбут А.А.,Финкельштейн Ю.Ю. Дискретное программирование. М.: Наука, 1969.