Лекція 8. Потоки на мережі

8.1. Постановка задачі.

Задачі про потік на мережі є більш загальними, ніж Т-задачі лінійного програмування.

Означення 8.1. *Орієнтованим графом* (*орграфом*) називається впорядкована пара $G=\{I,U\}$, де $I=\{i,j,...\}$ - непорожня множина вершин (вузлів), $U=\{(i,j):i,j\in I\}$ - множина впорядкованих пар, що називаються *дугами*, при цьому вершина i дуги (i,j) називається її *початком*, а j - її *кінцем*.

Геометрично орграф зображується точками (множина вершин I) та лініями зі стрілками (множина дуг U), що з‘єднують деякі пари цих точок.

Приміром, граф, для якого

$$I=\{1,2,3,4,5\},U=\{(1,2),(1,3),(1,4),(2,4),(4,3),(4,5),(5,2)\}$$

зображено на рис. 8.1.

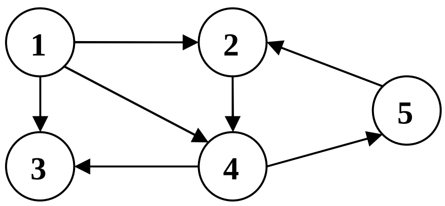


Рис. 8.1

Якщо орієнтація зв‘язку між вершинами є несуттєвою, то розглядають неорієнтовані графи.

Означення 8.2. *Неорієнтованим графом* називається впорядкована пара $G=\{I,U\}$, де I - множина вершин, $U=\{[i,j]:i,j\in I\}$ - множина невпорядкованих пар, що називаються *ребрами*.

Геометрично ребра зображують лініями, що з‘єднують відповідні вершини.

Вершини в графі називаються суміжними, якщо вони різні і існує дуга (ребро) , що їх з‘єднує.

Граф G називається *скінченним*, якщо скінченними є множини I та U.

Означення 8.3. *Шляхом* на графі G називається послідовність дуг $u_1,...,u_m\ \left(u_k=(i_k,j_k),k=\overline{1,m}\right)$, початок кожної з яких, починаючи з другої, співпадає з кінцем попередньої, тобто $i_{k+1}=j_k, k=\overline{1,m-1}$.

Шлях, що з‘єднує вершини i_1 та i_n можна задавати послідовністю вершин, через які він проходить.

Приміром, послідовність дуг $(1,2),(2,4),(4,5)$ формує шлях, що з‘єднує вершини 1 та 5 (рис. 8.1). Цей же шлях можна задати послідовністю вершин $(1,2,4,5)$.

Означення 8.4. *Ланцюгом* на графі G називається послідовність ребер $u_1,\dots,u_m$ $\left(u_k=[i_k,j_k],k=\overline{1,m}\right)$, в якій у кожного ребра, починаючи з другого, одна з вершин співпадає з однією з вершин попереднього, а друга – з якою-небудь вершиною наступного ребра.

Приміром, ребра $[1,2],[2,4],[4,5]$ утворюють ланцюг (див. рис.8.1), який можна задати послідовністю вершин $[1,2,4,5]$.

Якщо початкова вершина шляху (ланцюга) співпадає с кінцевою, то маємо *контур* (*цикл*).

Деревом називається зв‘язний граф, який не містить циклів (між довільними двома вузлами існує єдиний шлях).

Означення 8.5. *Мережею* називається граф, елементам якого поставлені у відповідність деякі параметри. *Елементами* графа вважають його вершини, дуги або більш складні конструкції, утворені з вказаних елементарних.

Побудуємо мережу наступним чином:

1) кожній вершині $i\in I$ поставимо у відповідність число d_i, що називається її *інтенсивністю*. Вершина i називається *джерелом*, якщо $d_i>0$, *стоком*, якщо $d_i<0$, і *нейтральною*, якщо $d_i=0$;

2) кожній дузі $(i,j)\in U$ поставимо у відповідність числа r_{ij} та c_{ij}, що називаються, відповідно, *функцією пропускної спроможності* та *функцією вартості*.

На практиці величини d_i часто інтерпретують як об‘єми виробництва ($d_i>0$) або споживання ($d_i<0$) деякого однорідного продукту в пункті (вершині) i. Величина r_{ij} визначає пропускну спроможність дуги (комунікації) (i,j), а величина c_{ij}, наприклад, собівартість транспортних перевезень вздовж дуги (i,j).

Потоком в мережі називається сукупність величин $X=\left\{x_{ij},(i,j)\in U\right\}$, що задовольняють умовам:

$$\sum_{j:(i,j)\in U}x_{ij}-\sum_{k:(k,i)\in U}x_{ki}=d_i,\ i\in I, \tag{8.1}$$

$$0 \le x_{ij} \le r_{ij}, (i,j) \in U. \tag{8.2}$$

Співвідношення (8.1) називаються *рівняннями збереження*, або *рівняннями неперервності*. Фізично вони означають, що різниця між величиною потоку, що виходить з вершини i та величиною потоку, що входить до неї, дорівнює її інтенсивності (рис. 8.2).

$$\sum_{k:(k,i)\in U} x_{ki} \quad \left.\begin{matrix}\to\\ \to\\ \to\end{matrix}\right\} \quad d_i \quad \left.\begin{matrix}\to\\ \to\\ \to\end{matrix}\right\} \quad \sum_{j:(i,j)\in U} x_{ij}$$

Рис. 8.2

Нехай $V \subset I$. Множина $U(V)$ дуг (i,j) таких, що $i \in V$, $j \notin V$, називається *розрізом мережі*, тобто $U(V) = \{(i,j) \in U : i \in V, j \notin V\}$.

Величина

$$r(V) = \sum_{(i,j)\in U: i\in V, j\notin V} r_{ij}$$

називається пропускною спроможність розрізу $U(V)$.

Загальні умови існування потоку на мережі встановлюють наступною теоремою.

Теорема 8.1. Для існування потоку на мережі, необхідно і достатньо, щоб

1) $d(I) = \sum_{i\in I} d_i = 0$;

2) для довільної множини $V \subset I$ виконувалась умова $d(V) = \sum_{i=1} d_i \le r(V)$.

Тобто, потік на мережі існує тільки тоді, коли сумарна інтенсивність усіх вершин мережі дорівнює нулю, а сумарна інтенсивність будь-якої підмножини вершин не перевищує пропускної спроможності розрізу мережі, що породжується цією підмножиною.

Кожному потокові $X = \{x_{ij}, (i,j) \in U\}$ поставимо у відповідність цільову функцію

$$Z(X) = \sum_{(i,j)\in U} c_{ij} x_{ij}.$$

Задача про оптимальний потік на мережі полягає у пошуку допустимого потоку X на мережі, що мінімізує цільову функцію $Z(X)$, тобто

$$Z(X)=\sum_{(i,j)\in U} c_{ij}x_{ij}\to \min,$$
$$\sum_{j:(i,j)\in U} x_{ij}-\sum_{k:(k,i)\in U} x_{ki}=d_i,\ i\in I, \qquad (8.3)$$
$$0\le x_{ij}\le r_{ij},\ (i,j)\in U.$$

Задача про оптимальний потік є частинним випадком ЗЛП, а отже, може бути розв'язана загальними методами лінійного програмування.

8.2. Задача про найкоротший шлях.

Розглянемо мережу, що визначається графом G, з означеною на U функцією вартості c_{ij}. Розглянемо також дві фіксовані вершини i_1, i_n графа G та довільний шлях, що з'єднує i_1 та i_n:

$$L=L(i_1,i_n)=\left((i_1,i_2),(i_2,i_3),\ldots,(i_{n-1},i_n)\right), \qquad (8.4)$$

де $i_k\in I,\ k=\overline{1,n}$.

Розглянемо функцію

$$C(L)=\sum_{k=1}^{n-1} c_{i_k i_{k+1}}, \qquad (8.5)$$

яка інтерпретується або як собівартість перевезення вантажу вздовж шляху L, або як довжина шляху L. В останньому випадку c_{ij} - довжина дуги $(i,j)\in U$.

Задача про найкоротший шлях полягає у пошуку шляху (8.4), що мінімізує цільову функцію (8.5):

$$C\left(L^*(i_1,i_n)\right)=\min_{L(i_1,i_n)} C\left(L(i_1,i_n)\right),$$

Шуканий шлях $L^*=\arg\min C(L)$ називається найкоротшим (оптимальним).

З формальної точки зору сформульована задача є частинним випадком задачі про оптимальний потік. Для цього розглядається мережа, вершина $s=i_1$ якої є джерелом одиничної інтенсивності, вершина $t=i_n$ - стоком одиничної інтенсивності, решта вершин – нейтральні.

Дугам приписуються необмежені пропускні спроможності $r_{ij}=\infty$, а собівартість перевезення по дузі дорівнює її довжині. Для потоку $X=\left\{x_{ij},(i,j)\in U\right\}$ у побудованій мережі

$$\sum_{j:(i,j)\in U} x_{ij} - \sum_{k:(k,i)\in U} x_{ki} = \begin{cases} 1, & i = s, \\ -1, & i = t, \\ 0, & i \neq s, t, i \in I, \end{cases}$$

$$x_{ij} \geq 0, (i, j) \in U.$$

Задача зводиться для знаходження потоку, що мінімізує цільову функцію

$$Z(X) = \sum_{(i,j)\in U} c_{ij} x_{ij}.$$

Легко зрозуміти, що для оптимального потоку $X^* = \{x^*_{ij}, (i,j) \in U\}$ величини x^*_{ij} будуть рівними 1 або 0 в залежності від того, входить чи ні дуга (i, j) до найкоротшого шляху. Тому $Z(X^*)$ порівнює довжині найкоротшого шляху, що з'єднує вершини i_1 та i_n. Для розв'язання задачі про найкоротший шлях розроблені методи, що враховують її специфіку.

ЗЛП для побудови дерева найкоротших шляхів має вигляд:

$$Z(X) = \sum_{(i,j)\in U} c_{ij} x_{ij} \to \min$$

$$\sum_{j:(i,j)\in U} x_{ij} - \sum_{k:(k,i)\in U} x_{ki} = \begin{cases} n-1, & i = s, \\ -1, & i \neq s, \end{cases}$$

$$x_{ij} \geq 0, (i, j) \in U.$$

8.3. Метод Мінті.

Одним з методів розв'язання задачі про найкоротший шлях є метод Мінті (метод позначок). Згідно цього методу процес розв'язування задачі складається із скінченного числа елементарних кроків, на кожному з яких позначаються вершини мережі та виділяються деякі її дуги.

Алгоритм методу Мінті.

Крок 1. Позначається вершина i_1 (коренева вершина) позначкою $h_{i_1} = 0, I^{(1)} = \{i_1(h_{i_1})\} = \{i_1(0)\}$ - множина позначених вершин. Нехай після виконання r кроків є множина $I^{(r)} = \{i_1(h_{i_1}), ..., i_k(h_{i_k}), ...\}$ позначених вершин i_k, кожній з яких поставлена у відповідність позначка h_{i_k}, що дорівнює найкоротшому шляху з вершини i_1 у вершину i_k.

Крок (r+1). 1. Будують розріз мережі, породжений множиною позначених вершин

$$U(I^{(r)}) = \{(i, j) \in U : i \in I^{(r)}, j \in I \setminus I^{(r)}\}.$$

2. Перевіряють умова $U\left(I^{(r)}\right)=\varnothing$. Якщо умова виконується, то кінець обчислень – дерево найкоротших шляхів побудовано. Його утворюють усі виділені дуги мережі. Якщо ні, то переходимо до наступного пункту.

3. Для кожної дуги $(i,j)\in U\left(I^{(r)}\right)$ обчислюють довжину шляху з початкової вершини i_1 в кінець цієї дуги h_i+c_{ij} та знаходять

$$\theta=\min_{(i,j)\in U\left(I^{(r)}\right)}\left(h_i+c_{ij}\right).$$

4. Виділяють усі дуги розрізу, для яких досягається θ. Якщо існує декілька таких дуг, що заходять у одну вершину, то вибирається тільки одна з них (довільна).

5. Кінці виділених дуг позначити числом θ та приєднати їх до множини $I^{(r)}$, утворюють множину $I^{(r+1)}$. Перейти до кроку $r+2$.

Кінець алгоритму.

За побудовою в кожній з позначених вершин (крім початкової) закінчується одна виділена дуга. Тому рух від вершини i_n вздовж виділених дуг у напрямку, протилежному їх орієнтації, реалізується однозначно. Крім того, початок виділеної дуги позначається раніше її кінця. Отже, послідовність виділених дуг, утворена в процесі руху від вершини i_n, утворює деякий шлях $L^*\left(i_1,i_n\right)$ з початком в вершині i_1 і кінцем у вершині i_n.

Теорема 8.2. Побудований методом Мінті шлях $L^*\left(i_1,i_n\right)$ є найкоротшим, що з‘єднує вершини i_1 та i_n, причому $C\left(L^*\left(i_1,i_n\right)\right)=h_{i_n}$.

$\triangle$ Скористаємось методом математичної індукції за номером кроку k, на якому позначена вершина i_n.

При $k=1$ маємо $i_n=i_1$ $h_{i_n}=h_{i_1}=0,$ $C\left(L^*\left(i_1,i_n\right)\right)=0\Rightarrow C\left(L^*\left(i_1,i_n\right)\right)=h_{i_n}=0$ і теорема справедлива.

Нехай твердження виконується для $k=r$. Це означає, що для довільної вершини $i_{n'}\in I^{(r)}$ шлях $L^*\left(i_1,i_{n'}\right)$ найкоротший і довжина його $C\left(L^*\left(i_1,i_{n'}\right)\right)=h_{i_{n'}}$.

Нехай вершина i_n позначена на $r+1$ -му кроці. Покажемо, що шлях $L^*\left(i_1,i_n\right)$, побудований за методом Мінті найкоротший з можливих. Запишемо його у вигляді:

$L^*(i_1,i_n)=\left\{L^*(i_1,i_{n'}),(i_{n'},i_n)\right\}$, де $i_{n'}\in I^{(r)}$ - позначена вершина з якої виходить виділена на $r+1$ -му кроці дуга $(i_{n'},i_n)$, що заходить у вершину i_n. Оскільки шлях $C\left(L^*(i_1,i_{n'})\right)$ найкоротший і його довжина $h_{i_{n'}}$, то

$$C\left(L^*(i_1,i_n)\right)=C\left(L^*(i_1,i_{n'})\right)+c_{i_{n'}i_n}=h_{i_{n'}}+c_{i_{n'}i_n}.$$

З іншого боку, за алгоритмом Мінті для позначки вершини i_n маємо

$$h_{i_n}=\theta=\min_{(i,j)\in U\left(I^{(r)}\right)}\left(h_i+c_{ij}\right)=h_{i_n}+c_{i_{n'}i_n},$$

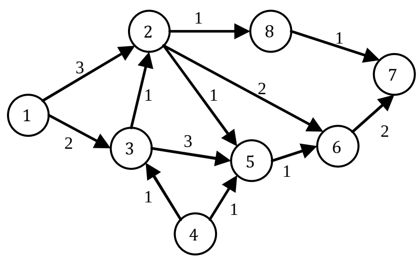
оскільки $(i_{n'},i_n)$ - виділена на $r+1$ -му кроці дуга.

Отже, теорема справедливий для усіх натуральних k .▲

Зауваження.

1. Сформульований алгоритм методу Мінті розрахований на знаходження єдиного найкоротшого шляху, що з'єднує вершини i_1 та i_n. Щоб знайти всю множину найкоротших шляхів, на кожному кроці слід виділяти всі дуги з мінімальним показником θ, навіть якщо декілька з них закінчуються в одній і тій самій вершині. А при зворотному русі з вершини i_n вздовж виділених дуг у кожній з вершин слід розглядати всі можливі розгалуження.
2. Насправді метод Мінті розв'язує більш загальну задачу: він знаходить найкоротший шлях з кореневої в кожну з позначених вершин.
3. Якщо процес позначення вершин методом Мінті продовжувати до тих пір, поки не припиниться розширення множини позначених вершин, то буде розв'язана задача найкоротшого шляху з початкової вершини i_1 у кожну позначену. Якщо при цьому деяка вершина i_λ мережі не попала до числа позначених, то немає шляху з i_1 в i_λ.

Приклад 8.1. Побудувати методом Мінті дерево найкоротших шляхів з коренем 1(s) на мережі.



Крок 1. Позначимо корінь дерева (вершину 1) (рис.8.3)

$h_1 = 0,\ I^1 = \{1\}$

Крок 2. Будуємо розріз мережі дугами $(1,2),(1,3)$, що виходять з множини I^1 і обчислюємо

$h_1 + c_{12} = 0 + 3 = 3$

$h_1 + c_{13} = 0 + 2 = 2$

Мінімальна з цих величин відповідає дузі $(1,3)$, ця дуга виділяється, вершина $\{3\}$ позначається числом $2(h_3 = 2), I^2 = \{1(0),3(2)\}$

Крок 3. Розглядаються дуги

$(1,2)\quad h_1 + c_{12} = 3$

$(3,2)\quad h_3 + c_{32} = 3$

$(3,5)\quad h_3 + c_{35} = 5$

Мінімальна з обчислених величин відповідає дугам $(1,2),(3,2)$. Виділяємо одну з них $(1,2),\ h_2 = 3, I^3 = \{1(0),2(3),3(2)\}$

Крок 4.

$(2,5)\quad h_2 + c_{25} = 4$

$(2,6)\quad h_2 + c_{26} = 5$

$(2,8)\quad h_2 + c_{28} = 4$

$(3,5)\quad h_3 + c_{35} = 5$

Виділяємо дуги $(2,5),(2,8), h_5 = 4, h_8 = 4, I^4 = \{1(0),2(3),3(2),5(4),8(4)\}$

Крок 5. Розглядаємо

$(2,6)\quad h_2 + c_{26} = 5$

$(5,6)\quad h_5 + c_{56} = 5$

$(8,7)\quad h_8 + c_{87} = 5$

Виділяємо дуги

$(5,6),(8,7), h_6 = 5, h_7 = 5, I^5 = \{1(0),2(3),3(2),5(4),6(5),7(5),8(4)\}$

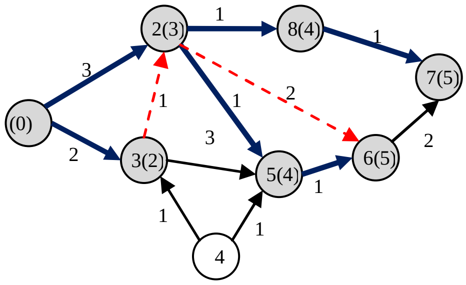


Рис. 8.3

Подальше позначення вершин неможливе. Повна множина позначених вершин $I^* = \{1,2,3,5,6,7,8\}$. Вершина 4 відсутня серед позначених, тобто немає шляху з вершини 1 у вершину 4.

Приміром, найкоротший шлях з вершини 1 у вершину 6 $L^*(1,6) = (1,2,5,6)$, який знаходимо переглядом виділених дуг від вершини 6 до вершини 1. При цьому, $C\left(L^*(1,6)\right) = h_6 = 5$.