

УДК 65.012.123(045)

АЛГОРИТМИ РОЗРАХУНКУ МЕРЕЖЕВИХ ГРАФІКІВ, ЗАДАНИХ СІМЕЙСТВОМ ВУЗЛОВИХ ПІДМНОЖИН

Литвиненко О. Є., д-р техн. наук, проф., *Мельник С. В.*, *Ментю М. О.*

Національний авіаційний університет

Litvinen@nau.edu.ua

Доведено переваги спискової форми представлення орієнтованого графа порівняно з матричною формою. Наведено алгоритми розрахунку мережних графіків, заданих сімейством вузлових підмножин.

Ключові слова: мережевий графік, вузлові підмножини, алгоритми розрахунку, орієнтований граф.

Advantages of the list form of the description of the oriented graph in comparison with the matrix form are proved. Algorithms of calculation of the network schedules set by family of nodal subsets are resulted.

Keywords: network schedule, subset of nodes, calculation algorithms, oriented graph.

Вступ

Мережеві графіки на практиці використовуються під час планування комплексів взаємозв'язаних робіт. Зокрема, в авіаційній сфері вони знайшли широке застосування при оперативному плануванні технологічних процесів усіх видів технічного обслуговування повітряних суден (ПС).

З формального погляду мережевий графік являє собою зважений орієнтований граф, дугам якого відповідають тривалості виконання окремих операцій (робіт), із яких складається розглядуваний технологічний процес. Структура такого графа відображає склад і логічну послідовність виконання необхідних робіт. Результатом розрахунку мережевого графіка є визначення критичного шляху, що характеризує тривалість виконання всього комплексу взаємозв'язаних робіт, та резерву часу кожної операції.

Існує два способи формального опису орієнтованих графів: матричний і списковий [1; 2]. Перший з них передбачає задання графа у вигляді однієї із матриць: суміжності, інцидентності, шляхів чи контурів. Списковий спосіб полягає в представленні орієнтованого графа сімейством вузлових підмножин, тобто сукупністю підмножин номерів дуг, інцидентних кожній його вершині. Матричний спосіб більш зручний з погляду алгоритмізації та програмування різноманітних задач, що розв'язуються на основі графів. Саме тому він широко викорис-

зання задач, що має вагоме значення для авіаційних систем оперативного управління технологічними процесами, функціонуючими в режимі реального часу. Цим пояснюється доцільність розробки методів розрахунку мережевих графіків, заданих списковим способом, незважаючи на підвищену (порівняно із матричним способом) складність відповідних алгоритмів.

Постановка проблеми

Нехай структура мережевого графіка формально представляється (як орієнтований граф) сімейством вузлових підмножин:

$$G = \{R_u^H, R_u^K \mid u \in U\},$$

де u — поточний номер вузла; U — множина вузлів (вершин, подій); R_u^H та R_u^K — множини номерів робіт, що починаються та завершуються в u -му вузлі відповідно.

Окрім структури мережевого графіка, будь-який технологічний процес характеризується такими параметрами:

t^a — заданий час завершення виконання всього комплексу робіт;

τ_r — нормативна тривалість виконання r -ї роботи; $r \in R$;

R — множина номерів робіт, що утворюють розглядуваний технологічний процес;

r — поточний номер роботи.

Мета розрахунку мережевого графіка полягає в знаходженні:

— тривалості виконання всього комплексу робіт;

— найбільш ранніх і найбільш пізніх термінів початку та завершенні виконання кожної роботи;

— резерву часу кожної роботи (що задає межі «здвигу» тієї чи іншої роботи ліворуч уздовж осі часу за відсутності на якому-небудь часовому відрізку необхідних фахівців — виконавців робіт).

© О.Є. Литвиненко, С.В. Мельник, М.О. Ментю, 2010

товується під час комп'ютерній реалізації алгоритмів розрахунку мережевих моделей. Однак списковий спосіб є більш економним для обсягів пам'яті, необхідних для опису графа, оскільки не потребує зберігання великої кількості нулів, що неминуче входять до складу тієї чи іншої матриці. При великих розмірностях графа така економія пам'яті приводить до швидкого розв'я-

Перед початком розрахунків необхідно визначити початковий u_i та кінцевий u_e вузли мережевого графіка.

Початковий вузол формально визначається за ознакою $R_{u_i}^e = \emptyset$, а кінцевий — за $R_{u_e}^i = \emptyset$.

Шукані параметри мережевого графіка встановлюються в результаті послідовної реалізації двох алгоритмів:

- прямого розрахунку, призначеного для визначення найбільш ранніх термінів початку і завершення виконання робіт;

- зворотного розрахунку, що дає змогу встановлювати найбільш пізні терміни початку та завершення виконання робіт.

Алгоритм прямого розрахунку мережевого графіка

Алгоритм прямого розрахунку передбачає багаторазову (покрокову) реалізацію уніфікованого модуля, на вхід якого на кожному μ -му ($\mu = 1, 2, 3, \dots$) кроці подається підмножина номерів вузлів U_μ , що підлягають розгляду (сенс і формальне визначення U_μ будуть розкриті нижче).

Цей модуль передбачає послідовне виконання таких операцій (операції 1—6 виконуються в циклі за параметром $u \in U_\mu$).

1. Вибір із U_μ і фіксація чергового елемента (номери вузла) u ($u \in U_\mu$).

На першому кроці $U_1 = \{u_i\}$, отже, $u = u_i$.

2. Фіксація підмножини R_u^e номерів робіт, що завершуються у вузлі u .

На першому кроці ця операція не виконується.

3. Визначення найбільш раннього терміну настання події, що відповідає вузлу u :

$$T_u^\delta = \max\{t_r^{\delta}; r \in R_u^e\},$$

де t_r^{δ} — найбільш ранній термін завершення виконання r -ї роботи.

На першому кроці дана величина, що відноситься до початкового вузла u_i , приймається рівною нулю: $T_{u_i}^\delta = T_{u_i}^\delta = 0$.

4. Перевірка умови завершення обчислень.

На першому кроці ця операція не виконується.

Якщо $u = u_e$, то обчислювальний процес завершується.

При цьому величина $T_u^\delta = T_{u_e}^\delta$ розглядається як строк завершення виконання всього комплексу робіт T . В іншому випадку виконується наступна операція.

5. Фіксація підмножини R_u^i номерів робіт, що починаються у вузлі u .

6. Обчислення для всіх робіт $r \in R_u^i$ найбільш ранніх термінів початку ($t_r^{i\delta}$) і завершення ($t_r^{\delta\delta}$) їх виконання:

$$t_r^{i\delta} = T_u^\delta;$$

$$t_r^{\delta\delta} = T_u^\delta + \tau_r; r \in R_u^i.$$

Це остання операція в циклі по параметру $u \in U_\mu$.

7. Формування підмножини номерів робіт, для яких на даному кроці встановлені найбільш ранні терміни початку та завершення їх виконання:

$$R_\mu = \bigcup_{u \in U_\mu} R_u^i.$$

8. Формування (накопичувальним шляхом) підмножини номерів робіт, для яких найбільш ранні терміни початку та завершення їх виконання встановлені до кінця розглянутого кроку алгоритму:

$$R(\mu) = \bigcup_{\mu'=1}^{\mu} R_{\mu'}.$$

9. Визначення підмножини номерів вузлів, для яких до кінця розглянутого кроку алгоритму встановлено тимчасові параметри всіх вхідних в них робіт:

$$U(\mu) = \{u \in U : R_u^K \subseteq R(\mu)\}.$$

10. Виділення в $U(\mu)$ підмножини номерів вузлів, для яких ще не встановлені найбільш ранні терміни початку та завершення виконання робіт, що виходять з них (тобто вузлів, які можуть і повинні бути розглянуті на наступному $(\mu + 1)$ -му кроці алгоритму):

$$U_{\mu+1} = U(\mu) \setminus \bigcup_{\mu'=1}^{\mu} U_{\mu'}.$$

Алгоритм зворотного розрахунку мережевого графіка

Алгоритм зворотного розрахунку можна вважати дзеркальним відображенням алгоритму прямого розрахунку, оскільки відправною точкою в ньому є кінцевий вузол мережевого графіка і час настання відповідної йому події T , обчислений у результаті реалізації першого алгоритму.

Другий алгоритм, як і перший, також передбачає багаторазову (покрокову) реалізацію уніфікованого модуля, на вхід якого на кожному μ -му ($\mu = 1, 2, 3, \dots$) кроці подається підмножина номерів вузлів U_μ , що підлягають розгляданню (сенс і формальне визначення U_μ будуть розкриті нижче). Цей модуль передбачає послідовне

виконання таких операцій (операції 1—6 виконуються в циклі за параметром $u \in U_\mu$).

1. Вибір із U_μ і фіксація чергового елемента (номеру вузла) $u (u \in U_\mu)$. На першому кроці $U_1 = \{u_\epsilon\}$ і, отже, $u = u_\epsilon$.

2. Фіксація підмножини R_u^i номерів робіт, що починаються у вузлі u . На першому кроці ця операція не виконується.

3. Визначення найбільш пізнього строку настання події, відповідного вузлу u :

$$T_u^i = \min\{t_r^{ii}; r \in R_u^i\},$$

де t_r^{ii} — найбільш пізній термін початку виконання r -ї роботи.

На першому кроці ця величина, що відноситься до кінцевого вузла u_ϵ , приймається рівною $T_u^\delta = T_{u_\epsilon}^\delta$ терміну виконання всього комплексу робіт T .

4. Перевірка умови завершення обчислень.

На першому кроці ця операція не виконується.

Якщо $u = u_i$, то обчислювальний процес завершується. В іншому випадку виконується така операція.

5. Фіксація підмножини R_u^ϵ номерів робіт, що закінчуються у вузлі u .

6. Обчислення для всіх робіт $r \in R_u^\epsilon$ найбільш пізніх термінів початку (t_r^{ii}) і завершення ($t_r^{\epsilon i}$) їх виконання:

$$t_r^{\epsilon i} = T_u^i - \tau_r; t_r^{\epsilon i} = T_u^i; r \in R_u^\epsilon.$$

Це остання операція в циклі за параметром $u \in U_\mu$.

7. Формування підмножини номерів робіт, для яких на даному кроці встановлені найбільш пізні терміни початку і завершення їх виконання:

$$R_\mu = \bigcup_{u \in U_\mu} R_u^\epsilon.$$

8. Формування (накопичувальним шляхом) підмножини номерів робіт, для яких найбільш ранні терміни початку та завершення їх виконання встановлені до кінця розглянутого кроку алгоритму:

$$R(\mu) = \bigcup_{\mu'=1}^{\mu} R_{\mu'}.$$

9. Визначення підмножини номерів вузлів, для яких до кінця розглянутого кроку алгоритму встановлено тимчасові параметри всіх вихідних у них робіт:

$$U(\mu) = \{u \in U : R_u^H \subseteq R(\mu)\}.$$

10. Виділення в $U(\mu)$ підмножини номерів вузлів, для яких ще не встановлені найбільш пізні терміни початку та завершення виконання робіт, що входять до них (тобто вузлів, які можуть і повинні бути розглянуті на наступному $(\mu + 1)$ -му кроці алгоритму):

$$U_{\mu+1} = U(\mu) \setminus \bigcup_{\mu'=1}^{\mu} U_{\mu'}.$$

Обчислення резерву часу робіт

Встановлення найбільш ранніх і найбільш пізніх моментів початку та завершення виконання робіт дає змогу обчислити їх резерв часу:

$$\rho_r = t_r^{ii} - t_r^{\delta} = t_r^{\epsilon i} - t_r^{\delta}; r \in R.$$

Приклад розрахунку мережевого графіка

Нехай $U = \{1, 2, \dots, 14\}$, мережевий графік заданий сімейством вузлових підмножин з табл. 1.

Таблиця 1

Сімейство вузлових підмножин

$R_1^H = \{11, 13, 21, 31, 41\}$	$R_8^H = \{43\}$	$R_1^K = \emptyset$	$R_8^K = \{42\}$
$R_2^H = \{15\}$	$R_9^H = \{16\}$	$R_2^K = \{13\}$	$R_9^K = \{14\}$
$R_3^H = \{23\}$	$R_{10}^H = \{17\}$	$R_3^K = \{21\}$	$R_{10}^K = \{15, 16\}$
$R_4^H = \{12, 22, 24\}$	$R_{11}^H = \{18\}$	$R_4^K = \{11\}$	$R_{11}^K = \{17\}$
$R_5^H = \{42\}$	$R_{12}^H = \{44\}$	$R_5^K = \{41\}$	$R_{12}^K = \{43\}$
$R_6^H = \{14\}$	$R_{13}^H = \{19\}$	$R_6^K = \{12, 22, 23\}$	$R_{13}^K = \{18, 25, 44\}$
$R_7^H = \{25\}$	$R_{14}^H = \emptyset$	$R_7^K = \{24, 31\}$	$R_{14}^K = \{19\}$

Нехай нормативна тривалість виконання робіт така, як наведено у табл. 2.

Таблиця 2

Тривалість виконання робіт

Номер роботи, r	Тривалість виконання, τ_r	Номер роботи, r	Тривалість виконання, τ_r	Номер роботи, r	Тривалість виконання, τ_r	Номер роботи, r	Тривалість виконання, τ_r
11	2	16	8	22	8	41	8
12	8	17	1	23	6	42	2
13	22	18	1	24	10	43	18
14	5	19	1	25	1	44	1
15	1	21	4	31	12	-	-

Реалізація алгоритму прямого розрахунку мережевого графіку:

Крок 1

$$U_1 = \{u_H\} = \{1\};$$

$$u = 1; T_1^P = 0; R_1^H = \{11, 13, 21, 31, 41\};$$

$$t_{11}^{HP} = 0; t_{11}^{KP} = 0 + 2 = 2; t_{13}^{HP} = 0;$$

$$t_{13}^{KP} = 0 + 22 = 22; t_{21}^{HP} = 0;$$

$$t_{21}^{KP} = 0 + 4 = 4;$$

$$t_{31}^{HP} = 0; t_{31}^{KP} = 0 + 12 = 12; t_{41}^{HP} = 0;$$

$$t_{41}^{KP} = 0 + 6 = 6;$$

$$R_1 = \{11, 13, 21, 31, 41\};$$

$$U_2 = \{2, 3, 4, 5\}$$

Крок 2

$$U_2 = \{2, 3, 4, 5\};$$

$$u = 2; R_2^K = \{13\}; T_2^P = 22; R_2^H = \{15\};$$

$$t_{15}^{HP} = 22; t_{15}^{KP} = 22 + 1 = 23;$$

$$u = 3; R_3^K = \{21\}; T_3^P = 4; R_3^H = \{23\};$$

$$t_{23}^{HP} = 4; t_{23}^{KP} = 4 + 6 = 10;$$

$$u = 4; R_4^K = \{11\}; T_4^P = 2;$$

$$R_4^H = \{12, 22, 24\}; t_{12}^{HP} = 2; t_{12}^{KP} = 2 + 8 = 10;$$

$$t_{22}^{HP} = 2; t_{22}^{KP} = 2 + 8 = 10; t_{24}^{HP} = 2;$$

$$t_{24}^{KP} = 2 + 10 = 12;$$

$$u = 5; R_5^K = \{41\}; T_5^P = 6; R_5^H = \{42\};$$

$$t_{42}^{HP} = 6; t_{42}^{KP} = 6 + 2 = 8;$$

$$R_2 = \{12, 15, 22, 23, 24, 42\};$$

$$U_3 = \{6, 7, 8\}$$

Крок 3

$$U_3 = \{6, 7, 8\};$$

$$u = 6; R_6^K = \{12, 22, 23\};$$

$$T_6^P = \max\{10, 10, 10\} = 10; R_6^H = \{14\};$$

$$t_{14}^{HP} = 10; t_{14}^{KP} = 10 + 5 = 15;$$

$$u = 7; R_7^K = \{24, 31\};$$

$$T_7^P = \max\{12, 12\} = 12; R_7^H = \{25\};$$

$$t_{25}^{HP} = 12; t_{25}^{KP} = 12 + 1 = 13;$$

$$u = 8; R_8^K = \{42\}; T_8^P = 8;$$

$$R_8^H = \{43\};$$

$$t_{43}^{HP} = 8; t_{43}^{KP} = 8 + 18 = 26;$$

$$R_3 = \{14, 25, 43\}; U_4 = \{9, 12\}$$

Крок 4

$$U_4 = \{9, 12\}$$

$$u = 9; R_9^K = \{14\}; T_9^P = 15;$$

$$R_9^H = \{16\};$$

$$t_{16}^{HP} = 15; t_{16}^{KP} = 15 + 8 = 23;$$

$$u = 12; R_{12}^K = \{43\}; T_{12}^P = 26; R_{12}^H = \{44\};$$

$$t_{44}^{HP} = 26; t_{44}^{KP} = 26 + 1 = 27;$$

$$R_4 = \{16, 44\}; U_5 = \{10\}$$

Крок 5

$$U_5 = \{10\};$$

$$u = 10; R_{10}^K = \{15, 16\};$$

$$T_{10}^P = \max\{23, 23\} = 23; R_{10}^H = \{17\};$$

$$t_{17}^{HP} = 23; t_{17}^{KP} = 23 + 1 = 24;$$

$$R_5 = \{17\}; U_6 = \{11\}$$

Крок 6

$$U_6 = \{11\};$$

$$u = 11; R_{11}^K = \{17\}; T_{11}^P = 24; R_{11}^H = \{18\};$$

$$t_{18}^{HP} = 24; t_{18}^{KP} = 24 + 1 = 25;$$

$$R_6 = \{18\}; U_7 = \{13\};$$

Крок 7

$$U_7 = \{13\}; u = 13; R_{13}^K = \{18, 25, 44\};$$

$$T_{13}^P = \max\{25, 13, 27\} = 27; R_{13}^H = \{19\};$$

$$t_{19}^{HP} = 27; t_{19}^{KP} = 27 + 1 = 28;$$

$$R_7 = \{19\}; U_8 = \{14\}$$

Крок 8

$$U_8 = \{14\};$$

$$u = 14; R_{14}^K = \{19\};$$

$$T_{14}^P = 28.$$

Оскільки $u = u_{\epsilon} = 14$, то обчислювальний процес завершується.

Мінімальний термін завершення виконання всього комплексу робіт становить $T = T_{u_k}^P = T_{14}^P = 28$ одиниць часу.

Реалізація алгоритму зворотного розрахунку мережного графіка:

Крок 1

$$U_1 = \{u_k\} = \{14\};$$

$$u = 14; T_{14}^{\Pi} = 28; R_{14}^K = \{19\};$$

$$t_{19}^{HP} = 28 - 1 = 27; t_{19}^{KP} = 28;$$

$$R_1 = \{19\}; U_2 = \{13\}$$

Крок 2

$$U_2 = \{13\};$$

$$u = 13; T_{13}^{\Pi} = 27;$$

$$R_{13}^K = \{18, 25, 44\};$$

$$t_{18}^{HP} = 27 - 1 = 26;$$

$$t_{18}^{KP} = 27; t_{25}^{HP} = 27 - 1 = 26;$$

$$t_{25}^{KP} = 27; t_{44}^{HP} = 27 - 1 = 26;$$

$$t_{44}^{KP} = 27; R_2 = \{18, 25, 44\};$$

$$U_3 = \{7, 11, 12\}$$

Крок 3

$$U_3 = \{7, 11, 12\};$$

$$u = 7; T_7^{\Pi} = 26; R_7^K = \{24, 31\};$$

$$t_{24}^{HP} = 26 - 10 = 16; t_{24}^{KP} = 26;$$

$$t_{31}^{HP} = 26 - 12 = 14; t_{31}^{KP} = 26;$$

$$u = 11; T_{11}^{\Pi} = 26; R_{11}^K = \{17\};$$

$$t_{17}^{HP} = 26 - 1 = 25; t_{17}^{KP} = 26;$$

$$u = 12; T_{12}^{\Pi} = 26; R_{12}^K = \{43\};$$

$$t_{43}^{HP} = 26 - 18 = 8; t_{43}^{KP} = 26;$$

$$R_3 = \{17, 24, 31, 43\}; U_4 = \{8, 10\}$$

Крок 4

$$U_4 = \{8, 10\};$$

$$u = 8; T_8^{\Pi} = 8; R_8^K = \{42\};$$

$$t_{42}^{HP} = 8 - 2 = 6; t_{42}^{KP} = 8; u = 10;$$

$$T_{10}^{\Pi} = 25;$$

$$R_{10}^K = \{15, 16\};$$

$$t_{15}^{HP} = 25 - 1 = 24;$$

$$t_{15}^{KP} = 25;$$

$$t_{16}^{HP} = 25 - 8 = 17; t_{16}^{KP} = 25;$$

$$R_4 = \{15, 16, 42\};$$

$$U_5 = \{2, 5, 9\}$$

Крок 5.

$$U_5 = \{2, 5, 9\};$$

$$u = 2; T_2^{\Pi} = 24; R_2^K = \{13\};$$

$$t_{13}^{HP} = 24 - 22 = 2; t_{13}^{KP} = 24;$$

$$u = 5; T_5^{\Pi} = 6; R_5^K = \{41\};$$

$$t_{41}^{HP} = 6 - 6 = 0; t_{41}^{KP} = 6;$$

$$u = 9; T_9^{\Pi} = 17; R_9^K = \{14\};$$

$$t_{14}^{HP} = 17 - 5 = 12; t_{14}^{KP} = 17;$$

$$R_5 = \{13, 14, 41\}; U_5 = \{6\}$$

Крок 6

$$U_5 = \{6\};$$

$$u = 6; T_6^{\Pi} = 12;$$

$$R_6^K = \{12, 22, 23\};$$

$$t_{12}^{HP} = 12 - 8 = 4; t_{12}^{KP} = 12;$$

$$t_{22}^{HP} = 12 - 8 = 4; t_{22}^{KP} = 12;$$

$$t_{23}^{HP} = 12 - 6 = 6; t_{23}^{KP} = 12;$$

$$R_6 = \{12, 22, 23\}; U_7 = \{3, 4\}$$

Крок 7

$$U_7 = \{3, 4\};$$

$$u = 3; T_3^{\Pi} = 6; R_3^K = \{21\};$$

$$t_{21}^{HP} = 6 - 4 = 2; t_{21}^{KP} = 6;$$

$$u = 4; T_4^{\Pi} = \min\{4, 4, 16\} = 4;$$

$$R_4^K = \{11\};$$

$$t_{11}^{HP} = 4 - 2 = 2; t_{11}^{KP} = 4;$$

$$R_7 = \{11, 21\};$$

$$U_8 = \{1\};$$

Крок 8

$$U_8 = \{1\};$$

$$u = 1; T_1^{\Pi} = \min\{2, 2, 2, 14, 0\} = 0.$$

Оскільки $u = u_H = 1$, обчислювальний процес завершується.

Найбільш пізній термін початку виконання всього комплексу робіт становить $T_1^{\Pi} = T_{u_H}^{\Pi} = 0$ одиниць часу, що підтверджує правильність розрахунків.

Резерв часу для робіт наведено у табл. 3.

Резерви часу для робіт

Номер роботи, r	Резерв часу, ρ_r	Номер роботи, r	Резерв часу, ρ_r	Номер роботи, r	Резерв часу, ρ_r	Номер роботи, r	Резерв часу, ρ_r
11	2	16	2	22	2	41	0
12	2	17	2	23	2	42	0
13	2	18	2	24	14	43	0
14	2	19	0	25	14	44	0
15	2	21	2	31	14	-	-

Висновки

Орієнтований граф, розглянутий у наведеному прикладі, містить 14 вершин і 19 дуг.

Його структуру можна описати матрицею суміжності, що має розмірність $14 \times 14 = 196$ елементів або матрицею інцидентності з розмірністю $14 \times 19 = 266$ елементів, тоді як для його представлення в списковій формі досить ввести в пам'ять комп'ютера числовий масив, що складається з 40 елементів.

Це доводить переваги спискової форми опису графів порівняно з матричною формою, незважаючи на відносну складність алгоритмів розрахунку мережевих графіків.

Програмну реалізацію наведених алгоритмів здійснено в операційному середовищі *Windows* з використання мови програмування C#.

ЛІТЕРАТУРА

1. Асанов М. О. Дискретная математика: графы, матроиды, алгоритмы / М. О. Асанов, В. А. Баранский, В. В. Расин. — Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001. — 288 с.
2. Харарри Фрэнк. Теория графов / Фрэнк Харарри; пер. с англ. и предисл. В. П. Козырева; под ред. Г. П. Гаврилова. — Изд. 2-е. — М. : Едиториал, 2003. — 296 с.

Стаття надійшла до редакції 22.11.2010.