УДК 519.1

АЛГОРИТМ СИНТЕЗУ МІНІМАЛЬНОГО ПОКРИВАЮЧОГО ДЕРЕВА НА ДОВІЛЬНІЙ ГРУПІ ВЕРШИН ЗВ'ЯЗНОГО ГРАФА

к.т.н. О.В. Кузнецов, В.М. Бацамут
(подав д.ф.-м. н., проф. С.В. Смєляков)

Описується точна поліноміальна алгоритмічна процедура, яка на довільній множині вершин початкового однокомпонентного графа будує мінімальне покриваюче дерево з урахуванням можливого транзитивного замикання (ТЗ) через вершини, що не увійшли до множини.

Задача побудови мінімального покриваючого дерева на групі вершин графа виникає, коли необхідно оптимально з'єднати задану множину елементів мережевого об'єкту з мінімальними витратами коштів і засобів (відновлення зв'язності деградованої мережі зв'язку, прокладка шляхів між вибраними містами деякої території і т.д.).

На даний час, для побудови мінімальних покриваючих дерев $\widetilde{\mathbf{G}}'_{[+]}$, де "+" - уся вузлова основа графа **G**, використовують найбільш відомі підходи Пріма, Краскала [1] та інші. Дані алгоритми також цілком можливо застосовувати для пошуку мінімальних покриваючих дерев $\widetilde{\mathbf{G}}_{[\mathbf{K}]}$, де **K** – довільна група вершин початкового графа **G**, шляхом перевірки на кожному кроці наявності в будуємому дереві зв'язності між усіма вершинами $\mathbf{v_i} \subset \mathbf{K}$. Проведений аналіз цих алгоритмів на різних структурах свідчить, що використання їх у цьому випадку може дати в кінцевому результаті велику похибку. Справа в тому, що в них на кожному кроці для зв'язування необхідних вершин $\mathbf{v_i} \subset \mathbf{K}$, до структури шуканого дерева додається ребро мінімальної ваги з наступною перевіркою на наявність циклу. Загальна сума ваги доданих ребер може перевищувати вагу деякого ребра, вага якого більша ніж кожного з доданих, але через яке здійснюється оптимальне транзитивне замикання. Враховуючи вище викладене сформулюємо та докажемо наступну теорему.

Теорема. Нехай $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ - довільний зв'язний граф. Мінімальне покриваюче дерево $\widetilde{\mathbf{G}}_{[\mathbf{K}]} = (\mathbf{K}, \mathbf{T})$ на групі вершин $\mathbf{v_i} \subset \mathbf{K}$ графа **G**, де $\mathbf{K} \subseteq \mathbf{V}$, може бути отримано за рахунок додання до складу множини **K** деякої вершини $\mathbf{v_i} \notin \mathbf{K}$, якщо через неї здійснюється оптимальне транзитивне замикання вершин $\mathbf{v_i} \subset \mathbf{K}$.

Доведення. Очевидно, що мінімальним покриваючим деревом на множині вершин **K**, якщо $|\mathbf{K}| = \mathbf{2}$, є найкоротший шлях, який об'єднує ці дві вершини. У випадку, коли $\left|\mathbf{K}\right| > \mathbf{2}$ таких шляхів може бути безліч. Таким чином, для отримання зв'язності деяких вершин **s**, **d**, **t** до структури шуканого графа $\widetilde{\mathbf{G}}_{[\mathbf{s,d,t}]}$ можуть бути додані $(\mathbf{s}, \mathbf{d_1})$, $(\mathbf{d_1}, \mathbf{d_2})$, …, $(\mathbf{d_{n-1}}, \mathbf{d_n})$, $(\mathbf{d_n}, \mathbf{d})$ та $(\mathbf{s}, \mathbf{t_1})$, $(\mathbf{t_1}, \mathbf{t_2})$, …, $(\mathbf{t_{n-1}}, \mathbf{t_n})$, $(\mathbf{t_n}, \mathbf{t})$ ребра. Припустимо, що в структурі початкового **G** присутня деяка вершина $\mathbf{t_n} \notin \mathbf{K}$ і ребро $(\mathbf{t_n}, \mathbf{d})$, для якого справедлива наступна умова: $\mathbf{w_{t_n,d} > w_{s,d_1}}$; $\mathbf{w_{t_n,d} > w_{d_1,d_2}}$; ... ; $\mathbf{w_{t_n,d} > w_{d_n,d}}$ та $\mathbf{w_{t_n,d} < \left(w_{s,d_1} + \ldots + w_{d_{n-1},d_n} + w_{d_n,d} \right)}$. Отже, тільки враховуючи ребро $(\mathbf{t_n}, \mathbf{d})$, можна знайти невідоме дерево $\widetilde{\mathbf{G}}_{[\mathbf{s,d,t}]}$.

У даній ситуації суміжні ребра, вагові коефіцієнти яких стоять в дужках, можуть значно збільшувати загальну вагу $\widetilde{\mathbf{G}}_{[\mathbf{s,d,t}]}$. Таким чином, поставлену задачу треба вирішувати з урахуванням можливого додання до структури шуканого дерева $\widetilde{\mathbf{G}}_{[\mathbf{K}]}$ додаткових вершин $\mathbf{v_i} \subset \mathbf{K}$, вага транзитивного замикання через які забезпечить оптимум загальної ваги. Теорему доведено.

Наприклад, є деяка мережа зв'язку. Мінімальне покриваюче дерево $\widetilde{\mathbf{G}}_{[1,2,3,6,7]}$ початкового графа **G,** збудоване за алгоритмом Краскала, надано на рис.1 більш виділеними лініями. При цьому план дорівнює $\mathbf{P\left[X\right] = \left\{ \left(l1,l4\right), \left(l1,l8\right), \left(l1,l9\right), \left(l1,l5\right), \left(l4,l2\right), \left(l8,l3\right), \left(l9,l7\right), \left(l5,l6\right) \right\}}$, а відповідна функція загальної ваги $\mathbf{F = \sum_{(i,j)\in P[x]} w_{ij} = 28}$.

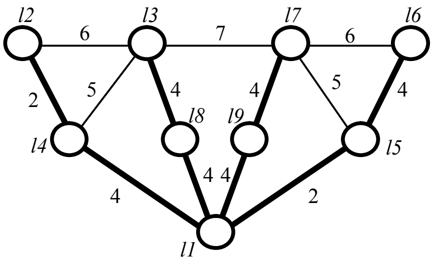

Рис.1. Покриваюче дерево $\widetilde{\mathbf{G}}_{[1,2,3,6,7]}$ (алгоритм Краскала)

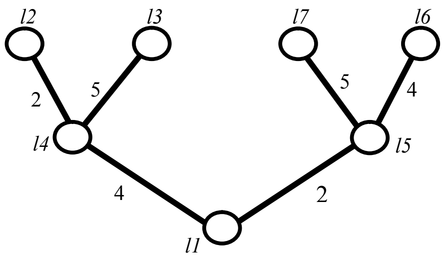

Рис. 2. $\widetilde{\mathbf{G}}_{[1,2,3,6,7]}$ – мінімальне покриваюче дерево

Але можна помітити, що оптимальне дерево $\widetilde{\mathbf{G}}_{[1,2,3,6,7]}$ складається з шести ребер плану $\mathbf{P\left[X\right] = \left\{ \left(l1,l4\right), \left(l1,l5\right), \left(l4,l2\right), \left(l4,l3\right), \left(l5,l7\right), \left(l5,l6\right) \right\}}$, при цьому $\mathbf{F = \sum_{(i,j)\in P[x]} w_{ij} = 22}$ (рис. 2). Абсолютна похибка складає 6 одиниць.

Алгоритм, що пропонується, засновано на заміні початкової структури графа $\mathbf{G}$ структурою $\widetilde{\mathbf{G}}'_{[+]}$, яка отримується шляхом введення до складу графа $\mathbf{G}$ всіх можливих транзитивних замикань, з корегуванням їх ваги між вершинами $\mathbf{v_i} \subset \mathbf{K}$, а саме

$$\mathbf{w}_{\langle \mathbf{i,j} \rangle_{\mathbf{T3}}} := \mathbf{w_{i,z}} + \mathbf{w_{z,j}}, \tag{1}$$

де $\mathbf{v_z} \not\in \mathbf{K}$ - вилучаєма разом зі своїми інцидентними ребрами, вершина з початкового $\mathbf{G}$.

Далі на отриманій структурі $\widetilde{\mathbf{G}}'_{[+]}$ одним із відомих алгоритмів будується мінімальне покриваюче дерево. Після цього по початковому $\mathbf{G}$ між суміжними вершинами збудованого дерева знаходиться найкоротший шлях з відповідною топологією.

Структура $\widetilde{\mathbf{G}}'_{[+]}$ отримується на матриці найкоротших шляхів $\mathbf{R_G}$ по матриці ваги початкового графа $\mathbf{G}$. Обчислення цього масиву надає інформацію про існування можливого ТЗ між всіма парами вершин, а також найменші значення цих зв'язків. При цьому, згідно (1), частина вершин $\mathbf{v_i} \not\subset \mathbf{K}$ буде вилучена з початкового $\mathbf{G}$, решта вершин $\mathbf{v_i} \not\subset \mathbf{K}$ залишиться і прийме участь у побудові структури $\widetilde{\mathbf{G}}'_{[+]}$. Оскільки мінімальним покриваючим деревом на множині вершин $\mathbf{K}$, якщо $|\mathbf{K}| = \mathbf{2}$, є найкоротший шлях, з'єднуючий їх, відзначимо важливе слідство теореми.

Слідство. Зменшити вагу транзитивного замикання вершин $\mathbf{v_i} \subset \mathbf{K}$ через деяку вершину $\mathbf{v_i} \not\in \mathbf{K}$, можна починаючи з $|\mathbf{K}| = 3$.

Представимо алгоритм у вигляді п'яти наступних кроків його роботи.

Крок №1. По матриці ваги початкового графа $\mathbf{G}$ обчислити його масив найкоротших шляхів $\mathbf{R_G}$.

Крок №2. По масиву $\mathbf{R_G}$ для кожної вершини $\mathbf{v_i} \not\in \mathbf{K}$ визначити три будь-яких вершини $\mathbf{v_i} \subset \mathbf{K}$, загальна вага суми найкоротших шляхів до яких мінімальна.

Результат операції: індекси стовпців (вершин $\mathbf{v_i} \subset \mathbf{K}$) *idx1*, *idx2*, *idx3*, мінімальна вага суми найкоротших шляхів - $\mathbf{w}^{\mathbf{v_i} \not\in \mathbf{K}}_{[\mathbf{v_{idx1}}, \mathbf{v_{idx2}}, \mathbf{v_{idx3}}]\mathbf{min}}$.

Крок №3. Для кожної вершини $\mathbf{v_i} \in \mathbf{K}$ за індексами, отриманими на кроці №2, обчислити відповідну вагу $\mathbf{w}^{\mathbf{v_i} \in \mathbf{K}}_{[\mathbf{v_{idx1}}, \mathbf{v_{idx2}}, \mathbf{v_{idx3}}]}$.

Крок №4. Вершину $\mathbf{v_i} \not\in \mathbf{K}$, для якої по всіх вершинах множини $\mathbf{K}$ виконується наступна нерівність, додати до складу множини $\mathbf{K}$:

$$\mathbf{w}^{\mathbf{v_i \notin K}}_{[\mathbf{v_{idx1}, v_{idx2}, v_{idx3}}]\mathbf{min}} < \mathbf{w}^{\mathbf{v_i \in K}}_{[\mathbf{v_{idx1}, v_{idx2}, v_{idx3}}]}. \qquad (2)$$

Вершини $\mathbf{v_i} \not\subset \mathbf{K}$, для яких нерівність (2) не виконується, вилучити з структури початкового $\mathbf{G}$. Якщо всі вершини $\mathbf{v_i} \not\subset \mathbf{K}$ проаналізовані – перейти до кроку №5, в противному випадку – до кроку №2.

Обґрунтування. Дійсно, якщо нерівність (2) справедлива, то є деяка вершина $\mathbf{v_i} \notin \mathbf{K}$, через яку здійснюється оптимальне, з точки зору загальної ваги, транзитивне замикання вершин множини $\mathbf{K}$, ніж без її урахування. Тому залучення її для побудови структури $\widetilde{\mathbf{G}}'_{[+]}$ призведе до поліпшення загального шуканого оптимуму. У разі вилучення будь-якої $\mathbf{v_i} \notin \mathbf{K}$ інформація про неї все рівно залишається в значенні найкоротших шляхів в масиві $\mathbf{R_G}$, якщо такі існують.

Крок №5: На отриманій множині вершин $\mathbf{K}$, згідно масиву $\mathbf{R_G}$, збудувати структуру $\widetilde{\mathbf{G}}'_{[+]}$ початкового графа $\mathbf{G}$.

Проілюструємо процедурність алгоритму на прикладі графа, наданого на рис. 1. Як і раніше, треба знайти мінімальне покриваюче дерево $\widetilde{\mathbf{G}}_{[1,2,3,6,7]}$. Обчислена матриця найкоротших шляхів $\mathbf{R_G}$ (3) має такий вигляд:

$$\mathbf{R_G} = \begin{array}{c|ccccccccc|c}
 & + & + & + & - & - & + & + & - & - & \\
 & \mathbf{l1} & \mathbf{l2} & \mathbf{l3} & \mathbf{l4} & \mathbf{l5} & \mathbf{l6} & \mathbf{l7} & \mathbf{l8} & \mathbf{l9} & \\
\hline
\mathbf{l1} & 0 & 6 & 8 & 4 & 2 & 6 & 7 & 4 & 4 & + \\
\mathbf{l2} & 6 & 0 & 6 & 2 & 8 & 12 & 13 & 10 & 10 & + \\
\mathbf{l3} & 8 & 6 & 0 & 5 & 10 & 13 & 7 & 4 & 11 & + \\
\mathbf{l4} & 4 & 2 & 5 & 0 & 6 & 10 & 11 & 8 & 8 & - \\
\mathbf{l5} & 2 & 8 & 10 & 6 & 0 & 4 & 5 & 6 & 6 & - \\
\mathbf{l6} & 6 & 12 & 13 & 10 & 4 & 0 & 6 & 10 & 10 & + \\
\mathbf{l7} & 7 & 13 & 7 & 11 & 5 & 6 & 0 & 11 & 4 & + \\
\mathbf{l8} & 4 & 10 & 4 & 8 & 6 & 10 & 11 & 0 & 8 & - \\
\mathbf{l9} & 4 & 10 & 11 & 8 & 6 & 10 & 4 & 8 & 0 & -
\end{array}. \qquad (3)$$

Знаком "+" позначені вершини $\mathbf{v_i} \subset \mathbf{K}$, а "-" – вершини $\mathbf{v_i} \not\subset \mathbf{K}$. Згідно кроку №2 серед вершин $\mathbf{v_i} \not\subset \mathbf{K}$ визначаємо $\mathbf{w}^{\mathbf{v_i \notin K}}_{[\mathbf{v_{idx1}, v_{idx2}, v_{idx3}}]\mathbf{min}}$. Для *l*4 це $\mathbf{w}^{\mathbf{l4 \notin K}}_{[\mathbf{l1, l2, l3}]\mathbf{min}} = \mathbf{11}$. Далі, згідно кроку №3, розраховуємо для кожної вершини $\mathbf{v_i} \in \mathbf{K}$ відповідну вагу $\mathbf{w}^{\mathbf{v_i}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]}$ і згідно кроку №4 зрівнюємо її з $\mathbf{w}^{\mathbf{l4}\notin\mathbf{K}}_{[\mathbf{l1,l2,l3}]\mathbf{min}}$. Так,

$$\mathbf{w}^{\mathbf{l1}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{14}; \quad \mathbf{w}^{\mathbf{l2}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{12}; \quad \mathbf{w}^{\mathbf{l3}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{14};$$

$$\mathbf{w}^{\mathbf{l6}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{31}; \quad \mathbf{w}^{\mathbf{l7}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{27}.$$

Згідно нерівності (2) вершина *l4* додається до множини $\mathbf{K}$. Для вершини *l5* відповідна вага $\mathbf{w}^{\mathbf{l5}\notin\mathbf{K}}_{[\mathbf{l1,l6,l7}]\mathbf{min}} = \mathbf{11}$. Для неї теж виконується нерівність (2) і вона також додається до множини $\mathbf{K}$. Для вершин *l8* та *l9* :

$$\mathbf{w}^{\mathbf{l8}\notin\mathbf{K}}_{[\mathbf{l1,l2,l3}]\mathbf{min}} = \mathbf{18}, \quad \mathbf{w}^{\mathbf{l9}\notin\mathbf{K}}_{[\mathbf{l1,l2,l7}]\mathbf{min}} = \mathbf{18}.$$

Для них нерівність (2) не виконується тому, що є $\mathbf{w}^{\mathbf{l1}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{14}$, $\mathbf{w}^{\mathbf{l2}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{12}$, $\mathbf{w}^{\mathbf{l3}\in\mathbf{K}}_{[\mathbf{l1,l2,l3}]} = \mathbf{14}$ та $\mathbf{w}^{\mathbf{l1}\in\mathbf{K}}_{[\mathbf{l1,l2,l7}]} = \mathbf{13}$ відповідно. Ці вершини з структури початкового $\mathbf{G}$ вилучаються, а в масиві $\mathbf{R_G}$ відповідні їм рядки та стовпці обнулюються. Таким чином матриця найкоротших шляхів $\mathbf{R_G}$ (4) графа $\tilde{\mathbf{G}}'_{[+]}$ має наступний вигляд:

$$\mathbf{R_G} = \begin{array}{c|ccccccccc} & l1 & l2 & l3 & l4 & l5 & l6 & l7 & l8 & l9 \\ \hline l1 & 0 & 6 & 8 & 4 & 2 & 6 & 7 & 0 & 0 \\ l2 & 6 & 0 & 6 & 2 & 8 & 12 & 13 & 0 & 0 \\ l3 & 8 & 6 & 0 & 5 & 10 & 13 & 7 & 0 & 0 \\ l4 & 4 & 2 & 5 & 0 & 6 & 10 & 11 & 0 & 0 \\ l5 & 2 & 8 & 10 & 6 & 0 & 4 & 5 & 0 & 0 \\ l6 & 6 & 12 & 13 & 10 & 4 & 0 & 6 & 0 & 0 \\ l7 & 7 & 13 & 7 & 11 & 5 & 6 & 0 & 0 & 0 \\ l8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}. \qquad (4)$$

Згідно кроку №5 по (4) збудуємо структуру $\tilde{\mathbf{G}}'_{[+]}$ (рис. 3). Застосовуючи до структури $\tilde{\mathbf{G}}'_{[+]}$ звісний алгоритм Краскала, а також ідентифікуючи в отриманому дереві по початковому графу $\mathbf{G}$ кожне його ребро між суміжними вершинами ребрами відповідного найкоротшого шляху (алгоритм Дейкстри), отримуємо шукане мінімальне покриваюче дерево $\tilde{\mathbf{G}}_{[1,2,3,6,7]}$, яке представлено на рис. 2.

Визначимо обчислювальну складність подібної алгоритмічної процедури. Дійсно, з початку на матриці ваги початкового графа $\mathbf{G}$ за допомогою відомого алгоритму Флойда будується масив найкоротших шляхів $\mathbf{R_G}$, обчислювальна складність якого оцінюється, як $\mathbf{O(N^3)}$ [2]. Далі, згідно кроків №№ 2,3,4, проводиться процедура побудови структури $\tilde{\mathbf{G}}'_{[+]}$, для чого потрібно $\mathbf{O(N^2)}$ обчислювальних витрат.

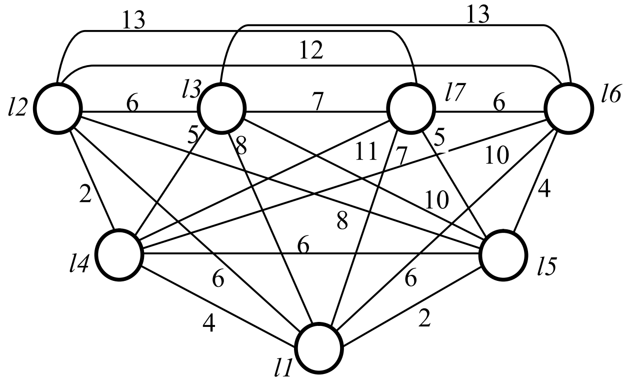

Рис. 3. Структура $\tilde{\mathbf{G}}'_{[+]}$ початкового графа $\mathbf{G}$

Для побудови мінімального покриваючого дерева та ідентифікації кожного його ребра потрібно також не більше $\mathbf{O(N^3)}$ витрат.

Таким чином, обчислювальну складність подібного алгоритму приблизно можна оцінити порядку $\mathbf{O(N^3)}$, що підтверджує його прикладну значимість у реальних розрахункових задачах.

ЛІТЕРАТУРА

1. *Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.*
2. *Липский В. Комбинаторика для программистов. – М.: Мир, 1988. – 213 с.*

Надійшла 14.01.2002

***КУЗНЕЦОВ Олександр Володимирович**, канд. техн. наук, доцент кафедри Харківського військового університету. Область наукових інтересів – дискретна математика, оптимізаційні задачі на графах.*

***БАЦАМУТ Володимир Миколайович**, ад'юнкт Харківського військового університету. Область наукових інтересів – оптимізація на мережевих об'єктах.*