УДК 519.8 doi: 10.15421/321919

В.А. Турчина, К.П. Гулько
Дніпровський національний університет імені Олеся Гончара

ЗАСТОСУВАННЯ АЛГОРИТМУ ФОРДА-ФАЛКЕРСОНА ДЛЯ ВИЯВЛЕННЯ НАДЛИШКОВОЇ ІНФОРМАЦІЇ

Для відомої задачі про максимальний потік розглянуто питання побудови або всіх максимальних потоків, або заданої їх кількості. Отриманні результати використовуються для дослідження питання існування надлишкової інформації в мережі. На основі класичного методу Форда-Фалкерона запропонована його модифікація та алгоритм для знаходження надлишкової інформації.

Ключові слова: сіткові мережі, алгоритм Форда-Фалкерсона, максимальний потік в мережі.

Для известной задачи о максимальном потоке рассмотрен вопрос построения всех максимальних потоков или заданного их количества. Полученные результаты используются для исследования вопроса существования избыточной информации в сети. На основании классического метода Форда-Фалкерсона предложена его модификация и алгоритм нахождения избыточной информации.

Ключевые слова: сетевые сети, алгоритм Форда-Фалкерсона, максимальный поток в сети.

The object of the article is the network maximum flow algorithm, mainly the Ford-Fulkerson algorithm. The algorithm began to be developed by two scientists: Ford and Fulkerson. This algorithm was proposed in order to find the maximum flow in the network. They began to be actively studied by scientists from the middle of the last century. The first report of "Maximum Network Flow" dates back to 1954. The authors of the report, Ford and Fulkerson had proved the theorem on the maximum flow and the minimum cross section for non-oriented graphs: the value of the maximum flow in the network is equal to the minimum throughput capacity of the section. The interest in the solution of these tasks was primarily due to practical needs, for that time construction of routes for the transportation of raw materials was not optimal and transported more raw materials than can transfer the connection between points. Such problems often arise when constructing connections that transport oil through pipes or transport coal through special excavators. The subject of the article is the problem of finding the maximum flow in the network. In graph theory, the transport network is an indicative graph in which each arc has no negative throughput and flow. Two peaks are distinguished: source and drain - such that any other vertex in the network lies on the way from source to drain.

The article consists of two sections. In the first section we consider the mathematical formulation of the problem and concrete examples of problems. The second section examines the classic Ford-Fulkerson algorithm, the modified Ford-Fulkerson algorithm to find excess information on the network, and the work of a modified algorithm on a specific example from the first section. The considered problems are relevant both from a theoretical and a practical point of view.

Keywords: Networks, Ford-Fulkerson algorithm, Maximum network flow.

Вступ. Серед задач дискретної оптимізації особливої уваги заслуговують оптимізаційні сіткові задачі, оскільки вони мають широке практичне застосування. В данній роботі розглядається одна із відомих сіткових задач, а саме задача про максимальний потік. Для її розв'язку відомий точний алгоритм поліноміальної складності для випадку цілочисельних спроможностей. В данній роботі вивчається задача аналізу сіткових мереж та потоків в них на предмет існування надлишкової інфориації, що дозволить суттєво економити ресурси.

Постановка задачі. Наведемо відому постановку задачі про максимальний потік.

Задано орієнтований зв'язний граф $G=\{V,U\}$, який складається з множин вузлів $V=\{1,2,3...n\}$ та множини дуг $U\subset V\times V$.

Кожній дузі (i, j) ставиться у відповідність невід'ємне число b_{ij}, яке називається пропускною спроможністю дуги. В мережі виділені два спеціальних вузла. Один з них, в який немає вхідних дуг, називається джерелом та позначається як s, другий, що немає вихідних дуг, називають стоком та позначають t. Таку мережу можно, наприклад, розглядати як водопровідну систему в якій труби відповідають дугам, джерело води відповідає джерелу s, стік води – t, а з'єднання – дуги в мережі. Пропускна спроможність дуги обмежує максимальний потік, який може проходити через дане сполучення. Потрібно визначити як саме пропустити максимальний потік з джерела в стік по цій мережі так, щоб мінімізувати втрати в цій мережі.

Сформулюємо цю задачу саме на математичній мові. Потоком з джерела s в стік t величини v в мережі називають множину невід'ємних чисел x_{ij} (кожне з яких визначається для дуги (i, j) в мережі), яка задовольняє наступним обмеженням:

$$\sum_i x_{ij}-\sum_j x_{jk}=\begin{cases}-v, & якщо\ \ j=s\\ 0, & якщо\ \ j\neq s,t\\ v, & якщо\ \ j=t\end{cases} \tag{1}$$

$$v\geq 0 \tag{2}$$

при обмеженнях на x_{ij}: $0\leq x_{ij}\leq b_{ij}$ для усіх (i, j).

Тут перша сума береться за дугами, які входять в вузол j, а друга сума – по дугам, які виходять з вузла j.

Обмеження (1) виражають, що в кожен вузол (окрім s та t) входить стільки потоку, скільки з нього виходить (закон збереження).

Нехай $X\subset U$, а $\bar{X}=U\setminus X$. Перерізом $(X,\bar{X})$ називається множина дуг (i, j) є U, для яких або $i\in X, j\in\bar{X}$, або $j\in X, i\in\bar{X}$ та видалення яких перетворює мережу в незв'язну. Переріз $(X,\bar{X})$ називається перерізом, який розділяє вузли s та t, якщо $s\in X, t\in\bar{X}$.

Пропускною спроможністю переріза $\left(X,\bar{X}\right)$ називається величина $C\left(X,\bar{X}\right)=\sum_{i,j} b_{ij}, i\in X, j\in \bar{X}$.

Існує відомий алгоритм для знаходження максимального потоку в мережі, а саме алгоритм Форда-Фалкерсона. Цей алгоритм має поліноміальну складність. Зрозуміло, що для тих дуг на яких потік дорівнює нулю, а також для дуг де він меньше пропускних спроможностей в деяких випадках можна економити ресурси. Виникає питання, а як визначити умови, при яких можна максимізувати цю економію. що максимальний потік може бути не єдиний.

В роботі запропоновано модифікацію алгоритму для знаходження усіх можливих потоків та процедуру знаходження надлишкової інформації в мережі.

Основна ідея алгоритму. Алгоритм базується на тому, що кроки при переході від однієї вершини до іншої запом'ятовуються. Модифікація полягає в тому, що пошук нового максимального потоку в s-t мережі враховує попередньо знайдені максимальні потоки. При кожному проході алгоритму ми порівнюємо теперішній шлях зі списком шляхів і якщо ми знайшли такий же, то перериваємо дію алгоритму та переходимо до розрахування надлишкової інформації в мережі та формулювання відповіді користувачу щодо зменшення пропускних спроможностей на деяких дугах.

Модифікація алгоритму. Розглянемо модифікований алгоритм Форда-Фалкерсона, що полягає в записуванні шляхів до списку для того, щоб знаходити усі можливі потоки або задану кількість, а також алгоритм для знаходження надлишкової інформації в мережі. Алгоритм базується на відомому алгоритмі Форда-Фалкерсона.

Крок 1 (процес розстановки поміток та запису їх).

В процесі роботи алгоритму кожен вузол знаходиться в одному з трьох станів: «не помічений», «помічений та не переглянутий» або «помічений та переглянутий». Спочатку усі вузли в мережі «не помічені». Помітка довільного вузла j завжди складається з двух частин. Перша частина – номер вузла i, котрий вказує, що можливо «надіслати» потік з i до j. Друга помітка – число, яке вказує максимальну величину потоку, котрий можливо «надіслати» з вузла i до вузлу j, не порушуючи обмежень на пропускну здатність дуг.

Також на першому кроці ми маємо пустий список дуг в котрий далі ми будемо заносити переглянуті дуги.

Джерело s завжди отримує фіктивну помітку $\left[s^{+},\varepsilon(s)=\infty\right]$.

Оберемо будь-який помічений та не переглянутий вузол j (на першому кроці це джерело s). Нехай він має помітку $\left[i^{+},\varepsilon(j)\right]$. З усіх вузлів, які суміжні з j, виділимо ті вузли k, які не помічені та для яких $x_{ik}<b_{ik}$. Припишемо кожному з вузлів k помітку $\left[j^{+},\varepsilon(k)\right]$, де $\varepsilon(k)=\min[\varepsilon(j),b_{jk}-x_{jk}]$ (такі вузли k тепер «помічені та не переглянуті»). Якщо вузол має помітку $\left[i^{-},\varepsilon(j)\right]$, то

після цього усім суміжним вузлам з j вузлам k, які не помічені і для котрих $x_{ik} > 0$, припишемо помітку $\left[j^{-}, \varepsilon(k) \right]$, де $\varepsilon(k) = \min[\varepsilon(j), x_{jk}]$ (такі вузли k тепер також «помічені та не переглянуті»). Тепер усі вузли, які є сусідами з j, мають помітки. Тоді вузол j вважається поміченим та переглянутим та його можливо більше не розглядати на цьому кроці. Може бути, що деякі сусідні вузли з j помічені, а інші не можуть бути поміченими (або усі сусідні з j вузли не можуть бути поміченими); в цих випадках вузол j також вважається «поміченим та переглянутим» та додається до списку вузлів. Знаки «+» та «-» в першій частині поміток вказують як повинен змінитися потік на кроці 2.

Якщо список дуг пустий, то ми не робимо звірку з ним і йдемо далі за алгоритмом, а якщо він містить дуги, то ми виключаємо з розгляду при першому проході перші три дуги з одного маршруту від s до t. Потім ми не ігноруємо їх, щоб знайти максимальний потік. Якщо на одному з кроків не вдається знайти мережу, яка відміна від знайдених, то алгоритм закінчує свою роботу та не переходимо на процедуру знаходження надлишкової інформації.

Продовжимо приписувати помітки вузлам, які є сусідніми до помічених та не переглянутих вузлів, до тих пір, поки або вузол t виявиться поміченим, або не можливо буде більше помітити жоден з вузлів та стік виявиться не поміченим. Якщо t не може бути поміченим, тоді не існує шляху з s до t, який збільшує потік, а отже побудований потік максимальний. Якщо t помічений, то на кроці 2 можливо знайти шлях, який збільшує потік.

Крок 2 (зміна потоку).

Припустимо, що стік t має помітку $[k^{+}, \varepsilon(t)]$. Тоді замінимо x_{kt} на $x_{kt} + \varepsilon(t)$. Якщо вузол k має помітку $\left[r^{-}, \varepsilon(k) \right]$, то замінимо x_{rk} на $x_{rk} - \varepsilon(k)$, а якщо $\left[r^{+}, \varepsilon(k) \right]$, то на $x_{rk} + \varepsilon(k)$. Записуємо дугу до початку списку з дугами. Продовжимо ці дії, поки не достигнемо джерела s. Після цього видалимо усі старі помітки вузлів і перейдемо до кроку 1.

Коли алгоритм закінчує роботу (на кроці 1 – не можемо помітити стік або ми отримуємо перші три дуги які вже існують в списку дуг, та які ми не розглядаємо), то отримуємо множину X помічених вузлів та множину дуг (i, j), де $i \in X$, $j \in \bar{X}$. Якщо список дуг не пустий, то ми заносимо його до списку маршрутів, які знайдені алгоритмом. Величина отриманого максимального потоку згідно з теоремою Форда-Фалкерсона буде дорівнювати пропускній спроможності отриманого перерізу $\left(X, \bar{X} \right)$.

Процедура знаходження надлишкової інформації в мережі. Нехай отримали множину максимальних потоків в s-t мережі. Для кожної з цих мереж на тих дугах, де потік не нульовий і менший за пропускні спроможності, зменшуємо пропускні спроможності на максимальну можливу величину. Обираємо ту мережу де досягнуто максимальне зменшеннях пропускних спроможностей.

Ілюстрація роботи на тестових прикладах. Проілюструємо роботу алгоритму на наступній мережі. Нехай кількість максимальних потоків дорівнює 3.

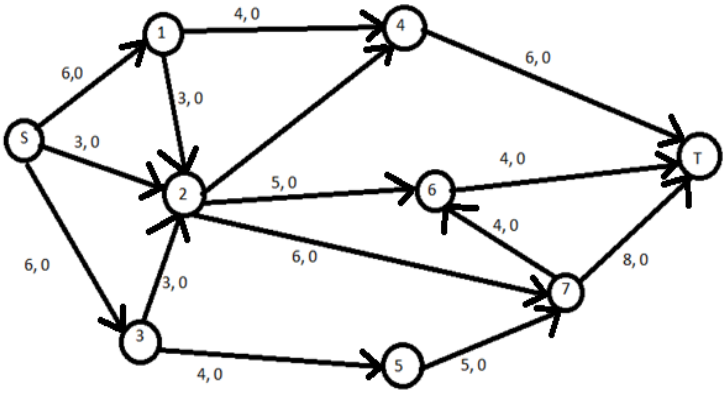

Рис. 1. Початкова мережа

Тепер будемо використовувати модифікований метод Форда-Фалкерсона для знаходження максимального потоку. Алгоритм повторюємо стільки разів стільки передбачено умовою задачі.

Перший прохід алгоритму:

1. S->2->6->T E(T) = 3. Збільшуємо на 3 одиниці.
2. S->1->4->T E(T) = 4. Збільшуємо на 4 одиниці.
3. S->1->2->4->T E(T) = 2. Збільшуємо на 2 одиниці.
4. S->3->5->7->T E(T) = 4. Збільшуємо на 4 одиниці.
5. S->3->2->6->T E(T) = 1. Збільшуємо на 1 одиницю.
6. S->3->2->7->T E(T) = 1. Збільшуємо на 1 одиницю.

C(X, $\overline{X}$) = 15.

Алгоритм закінчив свою роботу. Ми отримали наступну мережу:

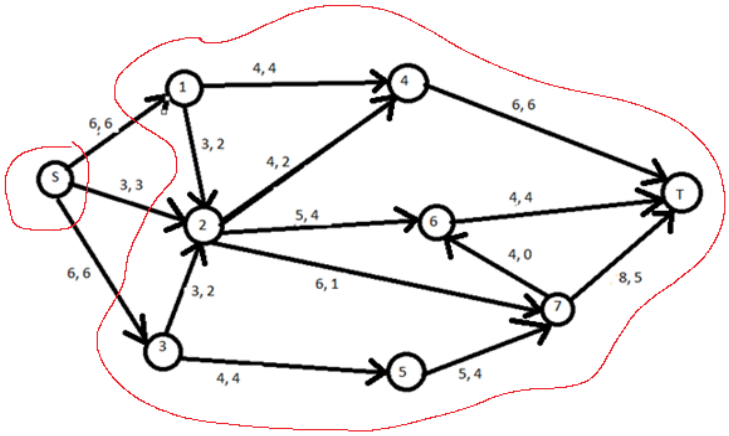

Рис. 2 Мережа після першої ітерації алгоритму

Другий прохід алгоритму:

1. S->2->7->6->T E(T) = 3. Збільшуємо на 3 одиниці.
2. S->3->2->7->6->T E(T) = 1. Збільшуємо на 1 одиницю.
3. S->3->5->7->T E(T) = 4. Збільшуємо на 4 одиниці.
4. S->3->2->7->T E(T) = 1. Збільшуємо на 1 одиницю.
5. S->1->4->T E(T) = 4. Збільшуємо на 4 одиниці.
6. S->1->2->7->T E(T) = 1. Збільшуємо на 1 одиницю.
7. S->1->2->6<-7->T E(T) = 1. Збільшуємо на 1 одиницю.

C(X, $\overline{X}$) = 15.

Алгоритм закінчив свою роботу. Ми отримали таку мережу:

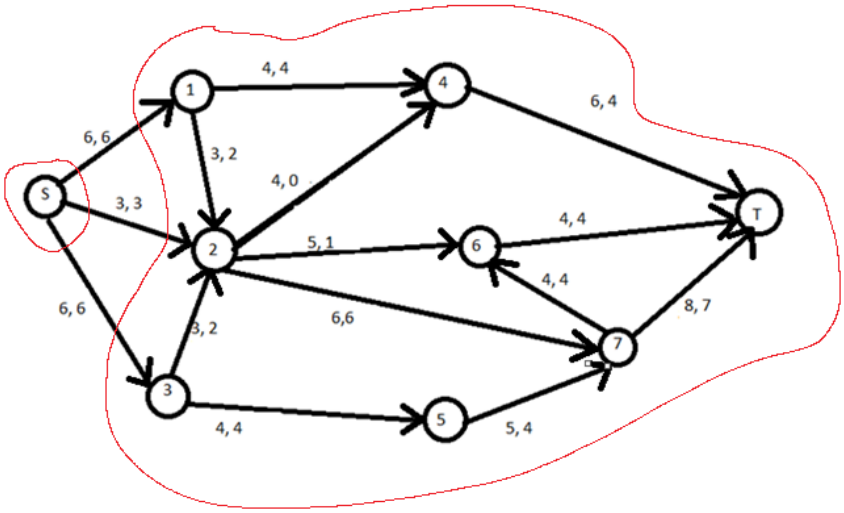

Рис. 3 Мережа після другої ітерації алгоритму

Третій прохід алгоритму:

1. S->2->7->T E(T) = 3. Збільшимо на 3 одиниці.
2. S->3->5->7->T E(T) = 4. Збільшимо на 4 одиниці.
3. S->3->2->7->T E(T) = 1. Збільшимо на 1 одиницю.
4. S->3->7->6->T E(T) = 1. Збільшимо на 1 одиницю.
5. S->1->4->T E(T) = 4. Збільшимо на 4 одиниці.
6. S->1->2->7->6->T E(T) = 1. Збільшимо на 1 одиницю.
7. S->1->2->6->T E(T) = 1. Збільшуємо на 1 одиницю.

C(X, $\overline{X}$) = 15.

Алгоритм закінчив свою роботу. Отримали таку мережу:

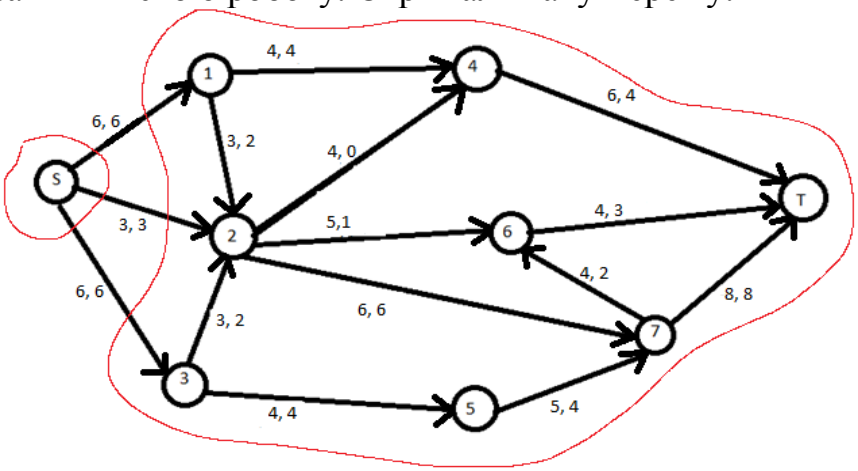

Рис. 4 Мережа після третьої ітерації алгоритму

Якщо ми подивимося на результат усіх проходів то ми можемо в деяких мережах зменшити пропускну спосібність.

З першого ми можемо зменшити:

- (1, 2) – 1
- (2, 4) – 2
- (2, 6) – 1
- (2, 7) – 5
- (3, 2) – 1

- (5, 7) – 1
- (7, 6) – 4
- (7, T) – 3

В цьому випадку можемо заощаджити 18 одиниць (труби або іншого матеріалу).

З другого:

- (1, 2) – 1
- (2, 4) – 4
- (2, 6) – 4
- (3, 2) – 1
- (5, 7) – 1
- (4, T) – 2
- (5, T) – 1
- (7, 6) – 1
- (7, T) – 1

В цьому випадку можемо заощадити 16 одиниць (труби або іншого матеріалу).

З трєтього:

- (1, 2) – 1
- (2, 4) – 4
- (2, 6) – 4
- (3, 2) – 1
- (5, 7) – 1
- (7, 6) – 2
- (4, T) – 1

В цьому випадку можемо заощадити 14 одиниць (труби або іншого матеріалу).

Алгоритм закінчив свою роботу. Аналіз отриманих результатів показує, що оптимальний економії ресурсів відповідає перша мережа.

Висновки. В роботі модифіковано алгоритм для знаходження заданої кількості або усіх можливих максимальних потоків в мережі та на основі їх аналізу даються рекомендації, щодо можливої економії ресурсів за рахунок зменшення пропускних спроможностей дуг.

Бібліографічні посилання

1. **Ермольев, Ю.М.** Экстремальные задачи на графах. [Текст] / Ю.М. Ермольев, И.М. Мельник. − К.: “Наукова думка”, 1968. − 176 с.
2. **Bozhenyuk, A.** The methods of maximum Flow and minimum cost flow finding in fuzzy network [Text] / A. Bozhenyuk, E. Gerasimenko, I Rozenberg // Proceedings of the Concept Discovery in Unstructured Data Workshop (CDUD 2012) co-located with the 10th International Conference on Formal Concept Analysis (ICFCA 2012) May 2012, Katholieke Universiteit Leuven, Leuven, Belgium 2012. – pp. 1-12.
3. **Форд, Л.** Потоки в сетях [Текст] / Л. Форд, Д. Фалкерсон.– М.: «МИР», 1966. – 276 с.

Надійшла до редколегії 26.09. 2019.