УДК 004.021; 519.1; 612.08; 681.5

Наталя МАНІЧЕВА, к.т.н., доцент,
Аделіна НІКОЛАЄВА, студент
Національний університет «Одеська політехніка», м. Одеса, Україна, e-mail: vmanichev@ukr.net, ad.nikolaieva@gmail.com

ВИКОРИСТАННЯ АЛГОРИТМУ ДЕЙКСТРИ В МЕДИЧНИХ МЕРЕЖЕВИХ ЗАВДАННЯХ

Анотація. В даній роботі розглянуто основні принципи роботи загальновідомого алгоритму Дейкстри, що призначений для пошуку оптимального рішення пошуку найкоротшого шляху за умов будь якої ситуації від мережі електричних ланцюгів до питань логістики.

Ключові слова: алгоритм Дейкстри, розподіл ресурсів, інформатика, мережа, медицина.

Завдання розподілу ресурсів виникають в найрізноманітніших областях науки, техніки та соціальної сфери. Характер розподілу ресурсів може бути різноманітним в залежності від розглянутої прикладної області та конкретного завдання.

В процесі вирішення завдання розподілу ресурсів павутини зв'язку можна виділити декілька основних етапів: виявлення вільних ресурсів, побудова найкоротшого маршруту, підключення абонента, перевірка дотримання всіх вимог, прийняття кінцевого рішення. На етапі побудови найкоротшого маршруту виникають складності з тим, щоб в випадку територіально розподіленої павутини зв'язку з великою кількістю вузлів та точок комунікації, мати високу вірогідність отримати оптимальне рішення без застосування для цього складної обчислювальної техніки.

Використання не раціональних маршрутів може призвести до непередбачених витрат, а також до складнощів і всіляких накладок у виконанні замовлень. Що спричинить погану репутацію та відсутність абонентів.

В наш час вирішення цього питання ще не доведено до автоматизму, а швидше відбувається на підставі суджень звичайного досвідченого працівника. Неминуча вплив суб'єктивності думки, що у вік інформаційних технологій мало прийнято.

Мережа зв'язку можна уявити графічною моделлю. Як вершин виступає безліч точок комутації, а як дуги – безліч ліній зв'язку. Приклад оного можна побачити на рис. 1.

Алгоритм закінчує роботу, коли викреслені всі вершини.

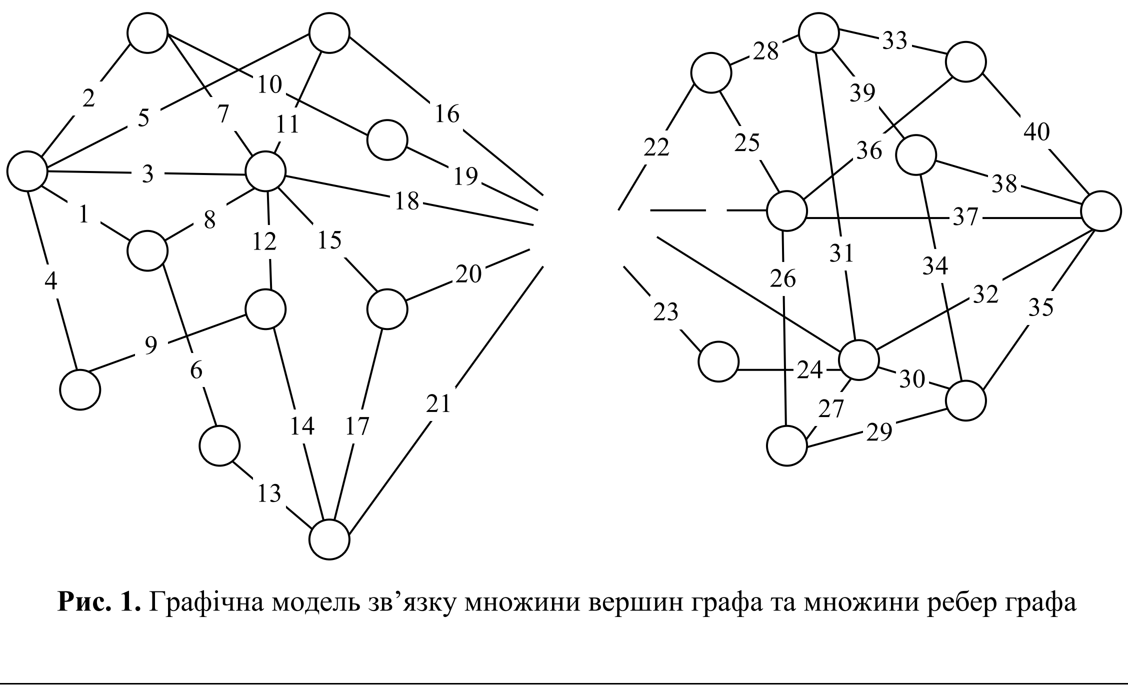

Рис. 1. Графічна модель зв'язку множини вершин графа та множини ребер графа

Алгоритм Дейкстри – не єдиний, але найпопулярніший з алгоритмів такого роду [1, с. 341 – 344]. Знаходить оптимальні маршрути та їх довжину між однією конкретною вершиною (джерелом) та рештою всіх вершин графа.

Інтуїтивно можна пояснити через такі пункти:

– ініціалізація;

– знаходимо вершину (із ще не опрацьованих), поточна найкоротша відстань до якої мінімальна;

– визначаємо довжину найкоротшої відстані;

– проводимо аналогічну операцію з сусідами першої обраної вершини;

– всі сусіди вершини 1 перевірені. Поточна мінімальна відстань до вершини 1 вважається остаточною і обговоренню не підлягає (те, що це дійсно так, вперше довів Дейкстра);

– знову знаходимо «найближчу» необроблену вершину;

– повторюємо те, що робили з першою обраною вершиною оминаючи вже оброблені;

– проробляємо те саме з вершинами, що залишилися.

Такий опис є дуже зжатим і не дуже зрозумілим для подальшого використання. Але цього достатньо для розуміння того, як воно взагалі працює. Більш детально можна ознайомитися з цим у праці самого Дейкстри [2, с. 269–271].

Класичний алгоритм Дейкстри працює тільки для графів без ребер від’ємної довжини. Може некоректно працювати, якщо граф має дуги негативної величини. Алгоритм Дейкстри не знаходить правильне рішення у випадку від’ємних ваг ребер (дуг) графа. Це призводить до ациклічних графів і найчастіше не можуть отримати право найкоротшому шляху.

Недоліком майже всіх алгоритмів знаходження найкоротшого шляху в графі і в тому числі алгоритму Дейкстри є те, що вони мають квадратичну і кубічну складності. Кількість шляхів експотенціально зростає , що при великій кількості ребер або дуг робить їх дуже повільними [3, с. 311].

Недоліком алгоритму Дейкстри є те, що не враховується важливість вузлів та ігноруються вузли, які знаходяться на значній відстані від цільової вершини. Для ліквідації цих недоліків необхідно якимось чином призначати пріоритети вузлам в залежності від відстані до цільової точки, щоб не кожній ітерації алгоритму ті вузли, які знаходяться далі від цілі, не розглядались [4, с. 425].

Малювання n графів або використання n матриць у процесі розрахунку найкоротших шляхів. Початкове призначення нескінченності довжини шляхів від входу до всіх вершин з наступним корегуванням їх значень. Це сповільнює роботу і ефективність алгоритму [5, с. 120].

Складність алгоритму Дейкстри залежить від способу знаходження вершини V, а також способу зберігання безлічі невідвіданих вершин і способи оновлення міток. Позначимо через n кількість вершин, а через m – кількість ребер у графі G.

У найпростішому випадку, коли для пошуку вершини з мінімальним $d[V]$ проглядається все безліч вершин, а для зберігання величин d – масив, час роботи алгоритму є $O(n^2+m)$. Основний цикл виконується порядку n раз, в кожному з них на знаходження мінімуму витрачається порядка n операцій, плюс кількість релаксацій (змін міток), яка не перевершує кількості ребер у вихідному графі.

Для розріджених графів (тобто таких, для яких m багато менше n^2). Невідвідані вершини можна зберігати в двійковій купі, а в якості ключа використовувати значення $d[i]$, тоді час вилучення вершини з U стане $\log n$, при тому, що час модифікації $d[i]$ зросте до $\log n$. Так як цикл виконується порядку n разів, а кількість релаксацій не більше m, швидкість роботи такої реалізації $O(n \log n + m \log m)$.

Якщо для зберігання невідвіданих вершин використовувати фібоначчієву купу, для якої видалення відбувається в середньому за $O(\log n)$, а зменшення значення в середньому за $O(1)$, то час роботи алгоритму складе $O(\log n + m)$ [6, с. 256].

Використовувати модель Дейкстри можливо різними способами. Творець використовував його для пошуку найкращого рішення для визначення оптимального шляху передачі електричного струму найбільш важливим елементам ланцюга, тим самим зводячи до мінімума розхід меді. Та широке застосування алгоритм отримав не лише у сферах інформатики та програмування. Його можна використовувати ще й при планування доріг та авіамаршрутів. В логістичних операціях. Також широко застосовують його й на благо технологій, як приклад – протоколи маршрутизації OSPF.

Але й це не кінець можливостей. Алгоритм можна застосовувати за-для потреб організації виставок (щоб відвідувач міг за один прохід подивитись усіх експонатів без повторювань), специфікації розміщення товару на полицях магазину (щоб звабити покупців на купівлю того, за чим вони не приходили).

Кожна людина або ж окремо взята фірма кожного дня вирішує проблеми на кшталт: як понизити витрати за умови обмежених ресурсів, як повисити ефективність продажу своїх послуг, як вибрати оптимальний курс розвитку. Дейкстра подбав про те, щоб ці та подібні питання було вирішити доволі просто (рис. 2).

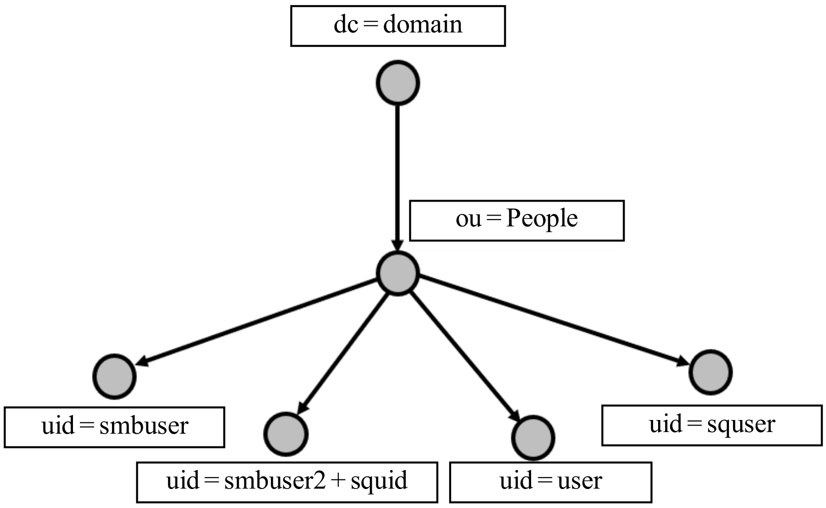

Рис. 2. Приклад стандартного дерева каталогу

Та як цей алгоритм може бути вигідно причеплений до питань медичної сфери? Дуже й дуже просто, б все зазначене вище може буди сказано й відносно потреб медицини: логістика препаратів між аптеками та лікарнями, виготовлення медичних пристроїв на базі програмування, оптимальний курс розвитку фармакологічної компанії, обмежені ресурси часної клініки… І ще більше прикладів можна навести, але думка ясна й без цього. Алгоритм Дейкстри в медичних мережевих завданнях використовується так само успішно як і в інших сферах.

Служба каталогу, як і більшість каталогів, має структуру організації зберігання даних у вигляді орієнтованого граф-дерева, де кожен "родоначальник" має один "нащадок". Ця структура детально описана у стандарті протоколу RCF 4512 та використовує ієрархічну модель даних.

Таку структуру можна описати графом $G = (V, E)$, де $V(G)$ – множина вершин графа $V(G) = (v\_d, v\_p, v\_0, \ldots v\_i)$, $E(G)$ – множина дуг графа, $E(G)$=($\{v\_d, v\_p\}, \{v\_p, v\_o\}$, … ... $\{v\_p, v\_i\}$).

Вершини зазначені згідно відносності до типу запису:

v_d – вершина кореневого елементу *dc*;

v_p – вершина-родоначальник для групи записів (ou = People);

v_i – вершина-нащадок для запису до каталогу, наприклад uid = mbuser;

i – rikmricnm pfgbcid, $i = 1, \ldots, n.$

Пошук записів у такому графі здійснюється по всьому дереву служби каталогів, отже час пошуку кожного запису прямо пропорційне та залежить від загальної кількості каталогу. Тут певно що не обходиться без вже добре нам відомого алгоритму. Найчастіше при настроюванні каталогу та сервісів, що взаємодіють з ним, вибирають одну гілку для зберігання та пошуку записів – ou = People. У такому разі час відповіді на запит до каталогу буде пов'язаний із вмістом гілки ou = People, а саме з кількістю записів у ній. Ця особливість пов'язана зі способом зберігання та отримання інформації у Berkeley.

Сучасна версія клієнтської та серверної частин LDAP дозволяє зберігати, записувати, здійснювати пошук запису в різних гілках каталогу, а сервіс-службам – працювати з ними. Групуючи записи за їхніми загальними параметрами (атрибутами та об'єктними класами, що використовуються сервісами), можна побудувати більш впорядковану структуру зберігання даних. Понад те, відстежуючи зміни у активності сервісів, можна організувати адаптивну службу каталогів, змінює свою структуру залежно від динаміки запитів до неї.

Література

1. Изотова Т.Ю. Обзор алгоритмов поиска кратчайшего пути в графе. Новые информационные технологии в автоматизированных системах, no. 19, 2016, pp. 341-344.

2. Dijkstra E.W. A note on two problems in connexion with graphs// Numerische Mathematatik. 1959. Vol. 1. P. 269-271.

3. Математичні моделі в менеджменті та маркетингу: Навч. посібник / [С.К. Рамазанов, Н.О. Рязанцева, Т.В. Ляшенко та ін.] – Луганськ : СПД Резніков В.С., 2010. – 311с.

4. Наконечний С.І. Математичне програмування: Навч. посіб. / С.І. Наконечний, С.С. Савіна – К. : КНЕУ, 2003. – 452 с.

5. Малиш Н.А. Моделювання економічних процесів ринкової економіки: Навч. пос. / Н.А. Малиш – К. : МАУП, 2004. – 120 с.

6. Сакович В.А. Исследование операций (детерминированные методы и модели): Справ. пособие / В.А. Сакович – Минск : Высшая школа, 1984. – 256 с.