УДК 004.424.36

Стьопкін А.В., Пластун Д.А.

[1] *старший викладач кафедри алгебри, ДВНЗ «ДДПУ»*

[2] *студент 4 курсу фізико-математичного факультету, ДВНЗ «ДДПУ»*

e-mail: stepkin.andrey@rambler.ru

АЛГОРИТМ ФОРДА-ФАЛКЕРСОНА

В роботі наведено реалізацію методу Форда-Фалкерсона для знаходження максимального потоку у транспортній мережі засобами мови програмування Pascal. Для реалізації методу Форда-Фалкерсона використовується метод обходу графа в глибину, що реалізований в мові програмування Pascal рекурсією.

Ключові слова: *рекурсія, обхід в глибину, метод Форда-Фалкерсона*

Вступ

В наш час інформаційні технології займають одне з найважливіших місць у всіх сферах нашого життя. Комп'ютери застосовуються скрізь: в навчанні, в менеджменті, в торгівлі, на виробництві та в інших видах діяльності людини. Але функціонування будь-якого комп'ютера неможливе без необхідних програм, а отже і алгоритмів, на основі яких пишуться програми. Таким чином, різноманітні алгоритми щодня допомагають людині у різних сферах діяльності. І деякі з них відіграють дуже важливу роль в розвитку людства. Отже, питання алгоритмізації є дуже актуальними й потребують багато уваги для подальшої розробки алгоритмів та вдосконалення вже існуючих. Поряд із цим, не може залишатися осторонь, також й процес програмування, як один з фундаментальних розділів інформатики.

Однією за важливих задач, вирішення якої допомагає оптимізувати транспортування вантажів, побудову нафто-, водо- та газопроводів, проектування електромереж є задача пошуку максимального потоку мережі. Для вирішення якої часто використовується метод Форда-Фалкерсона. Алгоритм реалізації якого, за допомогою методу обходу в глибину, ми і розглянемо більш детально.

Об'єктом дослідження є потоки в транспортних мережах.

Предметом дослідження є процес знаходження максимального потоку у транспортній мережі за допомогою алгоритму Форда-Фалкерсона.

Метою дослідження є реалізація алгоритму Форда-Фалкерсона, за допомогою методу обходу графа в глибину, мовою програмування Pascal.

Основна частина

Вперше проблеми пов'язані з транспортуванням потоків у мережах були розглянуті Канторовичем в 1933 році. Більше того — він розглядав більш загальну задачу з рухом потоку рідин різних типів. Основи теорії потоків були закладені в період з листопада 1954 по грудень 1955 дослідниками корпорації RAND (Санта-Моніка, Каліфорнія).

Перший звіт «Максимальний потік в мережі» датується 19 листопада 1954. Автори звіту — Форд і Фалкерсон, довели теорему про максимальний потік і мінімальний розріз для неорієнтованих графів: значення максимального потоку в мережі дорівнює мінімальній пропускній здатності розрізу. (Розрізом в мережі називається розбиття множини її вершин на два непересічних класи, таких що джерело і стік лежать у різних класах. Пропускною здатністю розрізу називається сума пропускних спроможностей ребер, кінці яких лежать в різних класах). Робота Форда і Фалкерсона про потоки і розрізи [5] була мотивована вивченням транспортних мереж залізниць. У тому ж звіті вони також описали простий алгоритм знаходження максимального потоку для планарних графів, що володіють такими додатковим властивостями: після додавання дуги з джерела в стік граф залишається планарним.

У своєму першому звіті про максимальні потоки, Форд і Фалкерсон згадали, що задача про максимальний потік була сформульована Харрісом наступним чином: "Розглянемо транспортну мережу залізних доріг, що з'єднують два міста через деяке число проміжних міст. Нехай також кожна дорога, що сполучає два міста, має деяку пропускну здатність. Знайти максимальний потік в даній мережі, враховуючи умову консервативності (тобто для будь-якого проміжного міста величина потоку, що прийшла в місто дорівнює величині потоку що вийшла з міста)". Пізніше у своїй книзі «Потоки в мережах» (1962), Форд і Фалкерсон дали більш точне посилання: в 1955 році Гарріс, у співавторстві з Россом, сформулювали просту модель для трафіку в транспортних мережах, і розглядали цю задачу (про мінімальний розріз). Мова йде про секретний звіт Харріса і Росса «Фундаментальний метод оцінки пропускних спроможностей транспортних мереж», датованому 24 жовтня 1955, і призначеному для ВПС США. На відміну від Форда і Фалкерсона, центральним завданням для Харріса і Росса було завдання про мінімальний розріз. А її застосуванням: знаходження слабких місць у системі залізниць СРСР.

(А саме, мінімальному розрізу тут відповідає мінімальний набір транспортних шляхів, знищення якого завдасть критичні пошкодження транспортному сполученню СРСР).

Основні поняття

Графом $G := (V, E)$ називається об'єкт, який заданий парою множин (V, E), де V — множина **вершин**, $E \subseteq V \times V$ — множина **ребер**. Граф називається скінченним, якщо множини його вершин і ребер є скінченними. Множину вершин графу G позначають $V(G)$, а множину ребер — $E(G)$.

Розглядають також орієнтовані графи.

Орієнтованим графом (орграфом) називається граф $G = (V, A)$, в якому (v_i, v_j) — впорядкована пара, де V — множина вершин , а елементи A називають *дугами* або **орієнтованими ребрами**.
Якщо $(v_i, v_j) = (v_j, v_i)$ то граф — **неорієнтований**, а елементи E називаються **ребрами**.

При зображенні орієнтованих графів напрямки ребер позначаються стрілками. Кожному неорієнтованому графу можна поставити у відповідність орієнтований граф з тією самою множиною вершин, в якій кожне ребро замінено двома орієнтованими ребрами, що є інцидентними тим самим вершинам і мають зворотні напрямки.

Граф $G = (V, E)$ називається **зваженим**, якщо його ребра (дуги) або вершини мають додаткові характеристичні ознаки (вагу).

Мережу можна представити як систему, яка транспортує деякий продукт з однієї точки в іншу. Цим продуктом можуть бути люди, електроенергія, природній газ, нафта та багато іншого. Використовуючи таке представлення розглянемо мережу як орієнтований граф, де кожному ребру $e = (v, u)$ відповідає додатне дійсне число $c(e)$, яке називається **пропускною спроможністю** ребра e. Якщо між вершинами немає ребра, то пропускна спроможність дорівнює нулю. Такий граф не може мати петель, адже розглядаються задачі для транспортування продукту тільки між різними вершинами. Необхідна умова: орієнтований граф повинен бути зв'язним, адже якщо є шлях з s в t, то нас будуть цікавити тільки компоненти, які містять s та t. Розглянемо також особливу вершину s, яка називається **джерелом**, — степінь заходу в неї дорівнює $\mathbf{0}$ $(d^+(S) = 0)$, а також вершину t, яка називається **стоком**, — степінь виходу з неї дорівнює також $\mathbf{0}$ $(d^-(T) = 0)$.

Мережа — це орієнтований граф $G = (V, E)$ разом з ваговою функцією $c : E \to R^+$ та виділеними вершинами s та t, такими що $d^+(s) = 0$ та $d^-(t) = 0$.

Для кожного ребра e розглянемо значення функції $f(e)$, яке визначає **потік** через це ребро. Зрозуміло, що величина потоку в ребрі не може перевищувати пропускну спроможність цього ребра. Також будемо вимагати, щоб потік, який заходить у вершину дорівнював потоку, який виходить з вершини.

Нехай S — підмножина множини вершин V та $T = V \setminus S$. Тоді множина $P = \{e : e \in (S, T)\}$ називається **розрізом**. Якщо $s \in S$ та $t \in T$, то розріз називається $s - t$ розрізом [1,2,4].

Знаходження максимального потоку у транспортній мережі за допомогою алгоритму Форда-Фалкерсона.

Ідея алгоритму [3] полягає в наступному. Ми вибираємо такий шлях від джерела до стоку, щоб для кожного ребра залишкова пропускна здатність була строго більше нуля. При цьому ребра на даному шляху можуть проходитися як у прямому, так і в зворотному напрямку. Вибираємо мінімальне значення серед залишкових пропускних спроможностей ребер даного шляху. Збільшуємо потік на кожному з ребер даного шляху на обране мінімальне значення. Далі шукаємо наступний аналогічний шлях. Робота алгоритму продовжується до тих пір, поки вдається знаходити дані шляхи. Відразу відзначимо, що даний алгоритм відноситься до класу недетермінованих, тобто кожен наступний крок алгоритму визначено неоднозначно. І час роботи (кількість кроків) алгоритму залежить від того, як будуть вибиратися кроки.

Алгоритм Форда-фалкерсона:

1. Прирівнюємо до нуля всі потоки. $\forall e(v_i, v_j) \in E \quad f(e) = 0$. Залишкова мережа спочатку збігається з вихідною мережею;
2. У залишкової мережі знаходимо будь-який шлях з джерела s у стік t. Дуги якого задовольняють умові $f(v_i, v_j) \leqslant c(v_i, v_j)$. Якщо такого шляху немає, то потік у мережі максимальний;
3. Пускаємо через знайдений шлях (він називається збільшувальним шляхом) максимально можливий потік;
4. На знайденому шляху в залишковій мережі шукаємо ребро з мінімальною пропускною здатністю C_{min};
5. Для кожного ребра на знайденому шляху збільшуємо потік на C_{min}, а в протилежному йому — зменшуємо на C_{min};
6. Модифікуємо залишкову мережу. Для всіх ребер на знайденому шляху, а також для протилежних їм ребер, обчислюємо нову пропускну здатність. Якщо нова пропускна здатність не дорівнює нулю, додаємо ребро до залишкової мережі, а якщо дорівнює нулю, стираємо його;
7. Повертаємося на крок 2.

Важливо те, що алгоритм не конкретизує, який саме шлях ми шукаємо на кроці 2 або як ми це робимо. З цієї причини алгоритм гарантовано сходиться тільки для цілих пропускних спроможностей, але навіть для них при великих значеннях пропускних спроможностей він може працювати дуже довго або зовсім не привести до оптимального рішення.

Реалізація алгоритму Форда-Фалкерсона мовою програмування Pascal.

Алгоритм Форда-Фалкерсона в паскалі можна реалізувати через пошук в глибину або через пошук в ширину. Ми реалізуємо цей алгоритм через пошук в глибину.

Вхідними даними буде масив який визначає пропускну здатність кожної дуги, якщо дуги не існує її пропускна здатність дорівнює нулю. Оскільки в транспортній мережі не може бути петель і дуги не можуть вести назад в джерело, та назад зі стоку то програма не вимагає їх вводу.

Результатом цієї програми буде число яке буде дорівнювати максимальному потоку в мережі.

```
program FordFulkerson;
    uses crt;
    const maxv=10; //кількість вершин графу
    var k,Cmin,Fmax,i,j:integer;
    graph: array [1..maxv, 1..maxv] of integer; //граф заданий масивом
    visited: array [1..maxv] of integer; //масив відвіданих вершин
    pathv: array[1..maxv] of integer; //масив вершин, що входять у
                                          збільшуваний шлях
    pathe: array[1..maxv-1] of integer; //масив пропускної спроможності ребер,
                                          що входять у збільшуваний шлях
procedure aff(v: integer);
    var i: integer;
    begin
      visited[v]:=1;
      pathv[k]:=v;
      for i := 1 to maxv do if (graph[v,i]>0) and (visited[i]=0) then //шукаємо
                                          вершини для збільшуваного шляху
      begin
        k:=k+1;
        aff(i);
      end;
    if v=1 then exit;
```

```
if (v>1) and (v<maxv) then
  begin
    for i:=1 to k-1 do visited[pathv[i]]:=0;
    visited[v]:=1;
    for i:= 1 to maxv do pathv[i]:=0;
    for i:=1 to maxv-1 do pathe[i]:=0;
    k:=1;
    v:=1;
    aff(1);
  end;
if v=maxv then
  begin
    for i:=1 to k-1 do pathe[i]:=graph[pathv[i],pathv[i+1]];
    Cmin:=pathe[1];
    for i:=1 to k-1 do if Cmin>pathe[i] then Cmin:=pathe[i];
    if Cmin=0 then exit;
    Fmax:=Fmax+Cmin;
    for i:=1 to k-1 do
      begin
        graph[pathv[i],pathv[i+1]]:=graph[pathv[i],pathv[i+1]]-Cmin;
        graph[pathv[i+1],pathv[i]]:=graph[pathv[i+1],pathv[i]]+Cmin;
      end;
    writeln;
    write('Відвідані вершини: ');
    for i:=1 to k do write(pathv[i],' ');
    writeln('Максимальний потік в даному шляху - ',Cmin);
    for i:=1 to maxv do
      begin
        graph[i,1]:=0;
        graph[maxv,i]:=0;
      end;
    writeln('Модифікований граф');
    for i:=1 to maxv do
      begin
        for j:=1 to maxv do write(graph[i,j], ' ');
        writeln;
      end;
    for i:=1 to maxv-1 do pathe[i]:=0;
```

```
        for i:=1 to maxv do pathv[i]:=0;
        k:=1;
        for i:=1 to maxv do visited[i]:=0;
        aff(1);
      end;
  end;
  procedure vvod;
    var i,j:integer;
    begin
      for i:=1 to maxv do for j:=1 to maxv do graph[i,j]:=0;
      for i:=1 to maxv-1 do
      for j:=2 to maxv do
      if i<>j then
        begin
          write('Введіть пропускну здатність дуги, 0 - якщо дуги немає
          (',i,',',j,') = ');
          readln(graph[i,j]);
        end;
      writeln('Початковий граф');
      for i:=1 to maxv do
        begin
          for j:=1 to maxv do write(graph[i,j], ' ');
          writeln;
        end;
    end;
  begin
    clrscr;
    k:=1;
    vvod;
    aff(1);
    writeln;
    writeln('Максимальний потік в мережі - ',Fmax);
    readln();
    end.
```

Висновки

В представленій роботі розглядався алгоритмом Форда-Фалкерсона та його використання на практиці. На основі цього алгоритму було складено програму пошуку максимального потоку в мережі. Отримані результати дослідження, зокрема розроблену програму можна використовувати для вивчення теорії алгоритмів, побудови програм з використанням рекурсії, також можна застосовувати програми для розрахунку максимального потоку в мережах. Робота буде корисна студентам при вивченні дискретної математики, або інформатики.

Література

1. *Вирт Н.* Алгоритмы и структуры данных. — М.: Мир, 1989. — 360 с.
2. *Касьянов В. Н., Евстигнеев В. А.* Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ-Петербург, 2003. — 1104 с.
3. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: Построение и анализ. — М.: МЦНМО, 2001. — 960 с.
4. *Редькин Н.П.* Дискретная математика. / Учебное пособие. — СПБ.: Лань, 2003. — 96 с.
5. *Форд Л.Р., Фалкерсон Д.Р.* Потоки в сетях. / Пер. с англ.. — М.: Мир, 1966. — 277 с.

Stepkin Andrey V., Plastun Dmytro A.
Donbas State Teachers' Training University, Slovians'k, Ukraine.

Ford–Fulkerson algorithm

Realization of Ford-Fulkerson method for finding maximum flow in transportational network by means of Pascal programming language is considered in the work. Depth-first search method, which is realized in Pascal programming language by a recursion is used for implementation of Ford-Fulkerson method.

Keywords: *recursion, depth-first search method, Ford-Fulkerson method.*
