ГОДИШНИК НА МИННО-ГЕОЛОЖКИЯ УНИВЕРСИТЕТ "СВ.ИВАН РИЛСКИ",Том 50,Св. III, Механизация,електрификация и автоматизация на мините, 2007
ANNUAL OF THE UNIVERSITY OF MINING AND GEOLOGY "ST. IVAN RILSKI", Vol. 50, Part III, Mechanization, electrification and automation in mines, 2007

ОПТИМАЛНИ ПОТОЦИ В ИНЖЕНЕРНИ МРЕЖИ

Петко Лалов[1], Стефан Димитров[2]

[1]Минно-геоложки университет "Св. Иван Рилски", 1700-София, e-mail: petko@mgu.bg
[2]Софийски университет "Св. Кл. Охридски", УИЦ, 1164-София, e-mail: stefan@ucc.uni-sofia.bg

РЕЗЮМЕ: Едни от най- важните задачи в инженерни мрежи са потоковите задачи. Редица реални ситуации в инженерната практика се свеждат до изследването на потоците от вход до изход в свързан граф, с приписана пропускателна способност на дъгите. Като примери могат да се посочат транспортния трафик, преноса на газ, въздух, вода и т.н. В тази връзка най- важната задача е намирането на максимален поток при ограничена пропускателна способност, която се решава с известният алгоритъм на Форд-Фалкерсон.
В тази работа сме си поставили за цел да дадем алгоритъм за генериране на минимален поток, при ограничения отдолу и двустранни ограничения, който може да се приложи при преносни мрежи.

OPTIMAL FLOWS IN ENGINEERING NETWORKS

Petko Lalov[1], Stefan Dimitrov[2]
[1]University of Mining and Geology "St. Ivan Rilski", 1700-Sofia, e-mail: petko@mgu.bg
[2]University of Sofia "St. Kl. Ohridski", UCC, 1164-Sofia, e-mail: stefan@ucc.uni-sofia.bg

ABSTRACT. Among the major tasks in engineering networks are the flow tasks. A number of real situations in engineering practice are reduced to the study of flows from input till output of a connected graph, with assigned throughput to the arcs. As instances can be pointed out transport traffic, transfer of gas, air, water, etc. In this respect, the most important task is to find out the maximum flow at limited throughput. The well known Ford-Fulkerson algorithm is applied in solving the latter. Constructing an algorithm to generate a minimum flow with limitations from below and with bilateral limitations which can be applied in transfer networks is the target set in the present work.

I. Въведение

Една от най- важните задачи от теорията на графите е намирането на максимален поток от вход до изход. При това на всяка дъга (x_i,x_j) е приписана пропускателна способност q_{ij}, която ограничава отгоре потока, протичащ по тази дъга. Така може да бъде оптимизирана интензивността на транспортния трафик между два пункта и редица още приложения могат да се намерят. За решаването на тази задача е известен алгоритъма на Форд-Фалкерсон.

В тази работа формулираме подобна задача: да се намери минималния поток от вход до изход в ориентиран граф, ако пропускателната способност на част от дъгите е ограничена отдолу, а на друга част е фиксирана. Такава задача възниква при регулиране на въздухоразпределението в РВМ. В този случай в някои клони на мрежата са зададени необходимите количества въздух(наложени дебити) и изисквания за минимални количества въздух в останалите клони. Търси се минималното общо количество въздух, което постъпвайки в мрежата ще удовлетвори поставените ограничения. Разбира се този проблем може да бъде третиран при всички преносни мрежи.

II. Математичен модел

Преносната мрежа е зададена чрез свързан ориентиран граф G(V,B) с вход s и изход t. С D означаваме множеството клони с фиксиран поток. В останалите клони (множество B-D) потока е ограничен отдолу. Ако клонът $(x,y) \in$ B-D и f(x,y) е потокът, протичащ по него, а c(x,y) е ограничението на потока отдолу, то търсим минималния поток от s в t при ограничения $f(x,y) \geq c(x,y)$, при $(x,y) \in$ B-D и f(x,y)=const при $(x,y) \in$ D.

Решаването на този проблем ще направим на два етапа: генерирането на начален поток, както и минимизиране на общия поток при зададен начален поток.

Нека имаме начален поток $f_0(x,y)$. За $(x,y) \in$ B-D $f_0(x,y)>c(x,y)$, а $f_0(x,y)$=const при $(x,y) \in$ D. Общият поток, постъпващ в s е Q_0. Целта е да намалим, доколкото е възможно Q_0. Ще въведем някои означения:

- дъга $(x,y) \in$ D ще причислим към тип N, ако в тази дъга потокът нито може да се увеличава, нито намалява;
- дъга $(x,y) \in$ B-D е от тип I, ако потокът в нея може да се увеличава(*увеличаваща се дъга*);
- дъга $(x,y) \in$ B-D е от тип R, ако потокът в нея може да се намали (*намаляваща дъга*).

Очевидно има дъги, които принадлежат и на двата типа I и R.

За да намалим общия поток Q_0 е необходимо да намерим верига от s до t, която да отговаря на едно от условията:

1. Веригата включва само дъги от тип R.

$$s \xrightarrow{R} a \xrightarrow{R} b \xrightarrow{R} c \xrightarrow{R} t$$

Ако i(x,y) е максималната величина, с която може да бъде намален потока в дъгата (x,y) от пътя, то min{ i(x,y) } дава количеството единици, с което може да бъде намален потока във всяка дъга от пътя, а оттам и Q_0.

2. Веригата включва дъги,както от I, така и от R, но дъгите от тип I са обратно ориентирани на дъгите от R.

```
     R          I        R       R
s ------> a <------ b ------> c ------> t
```

Потока в дъгите от R се намалява с min{ i(x,y) }, а в дъгите от I се увеличава с тази величина. С толкова се намалява и Q_0.

Проблемът е намирането на такива вериги. Потокът може да бъде намаляван, докато има в мрежата такива вериги. Затова ще дадем алгоритъм за тяхното търсене. В основата на този алгоритъм е построяване на дърво, състоящо се само от дъги от вида R и I.

Най-напред се определят множествата N, R и I. Дъгите от N се изключват от разглеждане. По нататък следват следните операции:

1. Оцветява се върха s.

2. Дъгата (x,y) или (y,x) и върха y,при вече оцветен връх x се оцветяват по следните правила:

- ако $(x,y) \in R$ оцветява се y и дъгата (x,y);

- ако $(y,x) \in I$ оцветява се y и дъгата (y,x);

- в останалите случаи оцветяване не се извършва;

В резултат на тази процедура върха t се оцветява, т.е. построили сме дърво, покриващо s и t и част от върховете. Следователно има верига от s до t, която може да намали общия поток. Тази верига ще наричаме *намаляща*. Ако не успеем да оцветим върха t, то такава верига не съществува. Изложения алгоритъм ще илюстрираме на следния пример(фиг.1):

1. оцветяваме върха s;

2. на оцветяване подлежат дъгите с начало s (s,a) и (s,b). Но $(s,a) \in I$ и не се оцветява. $(s,b) \in R$ и се оцветяват върха b и дъгата (s,b);

3. разглеждат се дъгите с начало или край b. Това са (a,b), (c,b) и (b,d); $(a,b) \in N$ и се изключва, $(c,b) \in I$ и се оцветява заедно с върха c. (b,d) и d също се оцветяват;

4. ако изберем за начало върха d, то инцидентните с него дъги (c,d) и (d,t) не могат да се оцветят и затова избираме върха c. От него може да се оцвети дъгата (c,t) и върха t, т.е. крайната цел е достигната;

В графа съществува верига (s,b), (c,b), (c,t), която може да намали общия поток.

Изложения алгоритъм е аналогичен на алгоритъма за търсене на увеличаваща верига(*аугментална*) на Форд-Фалкерсон и ще го наричаме FFR.

Сега можем накратко да опишем алгоритъма за минимален поток при ограничения отдолу:

1. избираме произволен начален поток, който отговаря на ограниченията. Той може да бъде получен в резултат на решението на някоя задача в мрежа, както и определен самостоятелно;

2. определяме множествата R, I и N. За всяка дъга (x,y) $\in$ B-D за която f(x,y)>c(x,y) изчисляваме i(x,y)=f(x,y)-c(x,y);

3. прилагаме алгоритъм FFR;

4. ако се намери намаляща верига от s до t, то общия поток се намалява с min{ i(x,y) }, където (x,y) е дъга от веригата. След това отново се изпълнява точка 2.

Ако при прилагане на алгоритъм FFR върха t не може да се оцвети, съществуващият поток е оптимален.

Остава да дадем алгоритъм за формиране на начален поток. Ще дефинираме някои понятия. В реалния случай преносната мрежа се представя посредством свързан ацикличен граф с един вход и един изход. Както е известно върховете на този граф могат да бъдат номерирани по такъв начин, че за всяка дъга (x,y), N(x)<N(y), където N(x) означава номера на върха x.

За всеки връх, различен от s и t имаме входящ и изходящ поток. В началото входящия поток за върха x е

$P_x^+ = \Sigma\, c(i,x)$, а изходящия е $P_x^- = \Sigma\, c(x,j)$

Върховете на графа разделяме на следните типове:

1. тип вход/изход-> s и t;

2. върхът x е от тип (+), ако $P_x^+ < P_x^-$;

3. върхът x е от тип (-), ако $P_x^+ > P_x^-$;

4. върхът x е от тип (=), ако $P_x^+ = P_x^-$.

Очевидно ние се стремим всички върхове без s и t да станат от вида (=). За всеки връх (+) трябва да се увеличи количеството единици, постъпващи в него, а за връх (-) трябва да се увеличат количеството единици, излизащи от него. При изравняването на потока в даден връх x, евентуално се променя типа на върховете y от дъгите (x,y) или (y,x). Очевидно, ако всички върхове са (=) и (-),то започвайки изравняването от връх (-) с най-малък номер, последователно изравняваме върховете (-) по нарастване на техните номера. Не е трудно да се прецени, че изравняването на връх x(-) води евентуално до промяна на типа само на върховете y с по-големи номера. При това, ако y е от тип (-) той ще запази типа си при изравняването на върха x. На фиг.2 се демонстрират тези постановки.

При изравняването на върха x ще се увеличи потока по дъгите (x,y_1) и (x,y_2). Очевидно върхът y_1 ще остане от тип (-), а върхът y_2 от (=) ще стане (-).

Изравняването на върхове от тип (+) води до промяна само на върхове, техни предшественици. Затова в предлагания алгоритъм всички върхове от тип (+) изравняваме по низходящ ред на техните номера. Ето и описанието на този алгоритъм:

1.номерираме върховете на графа G по правилото, ако (x,y) е дъга то N(x)<N(y).

2. определяме типа на върховете;

3. изравняваме върховете от тип (+), започвайки от този с най-голям номер и движейки се в низходящ ред. Изравняване се извършва така:ако x е изравнявания връх, то количеството $P_x^+ - P_x^-$ се прибавя към c(y,x) на дъгата (y,x), като при няколко върха y, се избира този с най-малък номер;

4. след прилагане на точка 3, върховете на G ще станат само от тип (-) и (=).Изравняването на връх x(-) започва от min N(x) и продължава във възходящ ред. За да се изравни потока в x, то към c(x,y) на дъгата (x,y) се прибавя количеството $P_x^+ - P_x^-$, като при няколко върха y се избира този с най-голям номер.

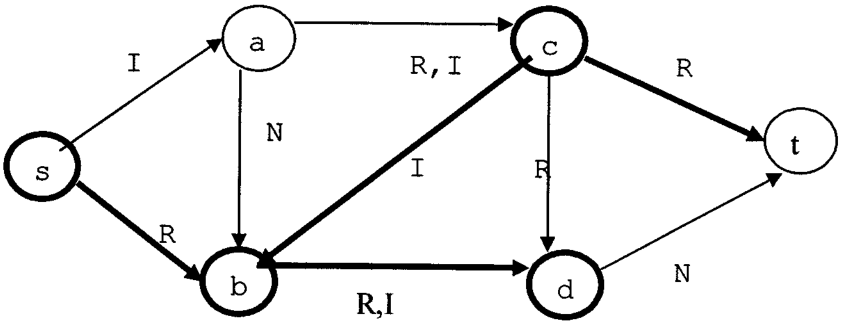
Ще илюстрираме алгоритъма с един пример(фиг.3).

Върховете от тип (+) са a, c и e. Започваме изравняването на тези върхове в ред e, c, a.

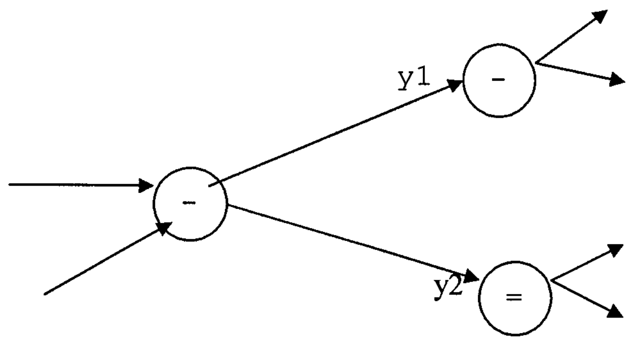
Изравняваме върховете (-) по растене на техните номера, т.е. в реда b, d, f.

По нататък прилагайки алгоритъма за търсене на намаляващи вериги можем да намалим началния поток. Това може да стане по веригата s - a - c - e - t с количество 5.

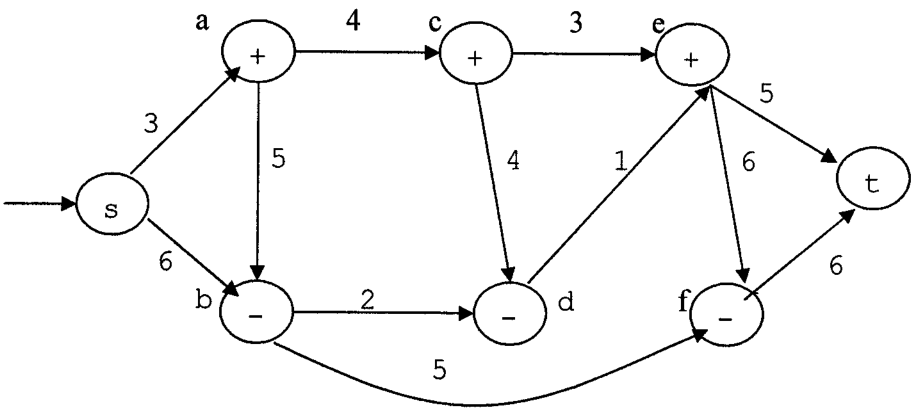
В този граф намаляща верига вече не съществува и това е оптималния поток.



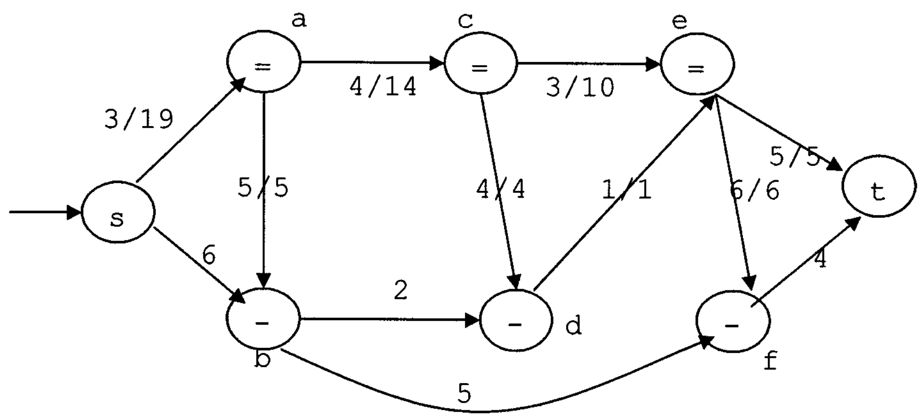
фиг.1



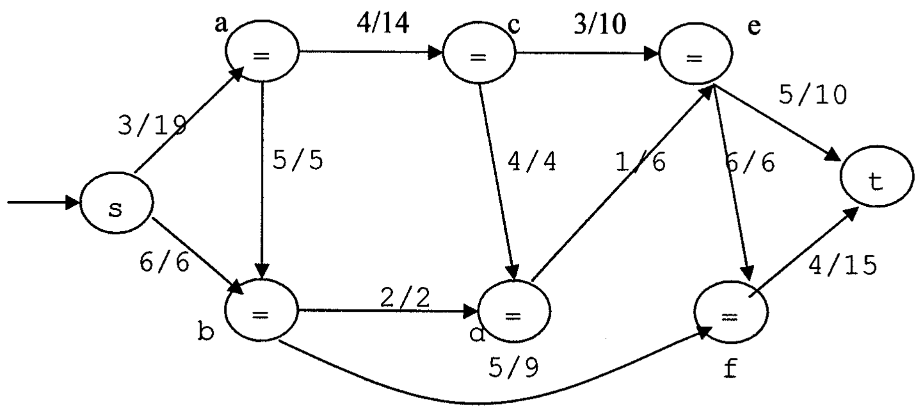
фиг.2



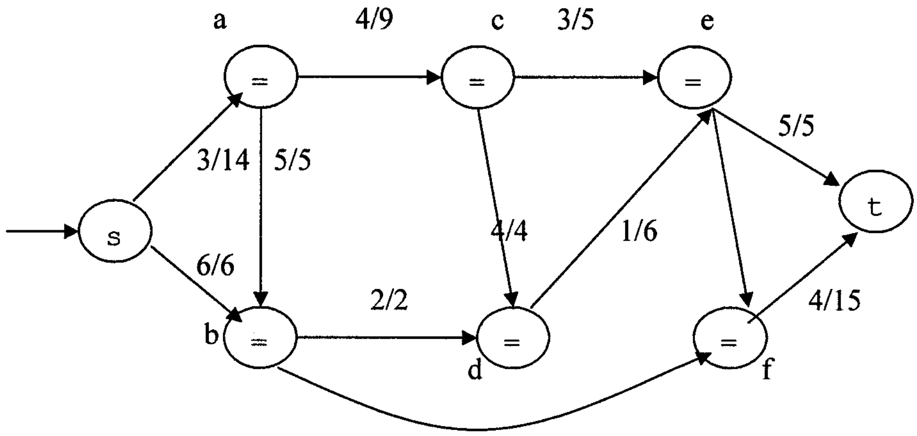
фиг.3



фиг.4



фиг.5



фиг.6

III. Заключение

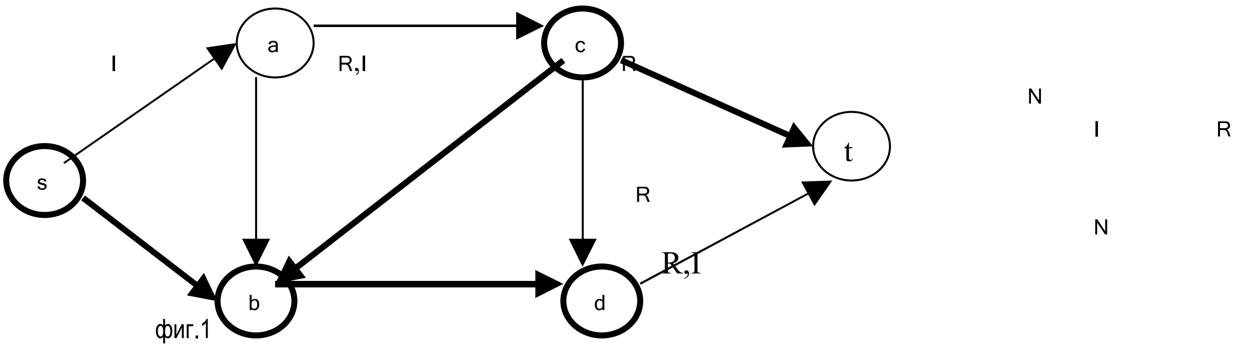
Претенциите ни в тази работа са в създаването на алгоритъм за търсене на минимален поток с ограничения отдолу. Основна роля в изложения алгоритъм играе ефективна процедура за генериране на начален допустим поток. В минно-руднична интерпретация този принос дава възможност да се определи минималното общо количество въздух, което трябва да постъпи в РВМ и удовлетвори поставени ограничения. Алгоритъмът е приложим за всички преносни мрежи.

Препоръчана за публикуване от
катедра "Информатика", МЕМФ

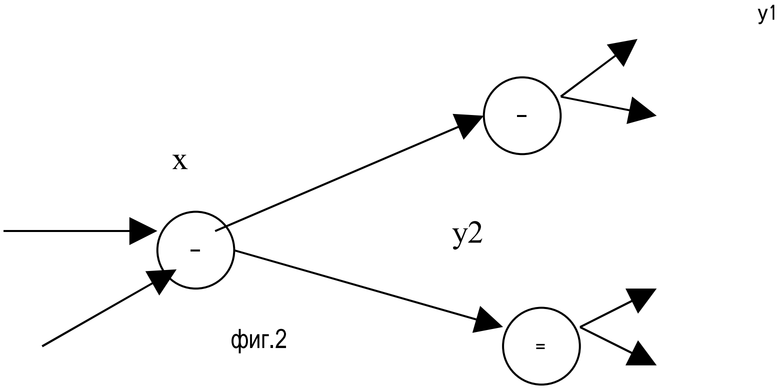
ЛИТЕРАТУРА

1. Р.Басакер, Т.Саати Конечыe графы и сети, Moscow, 1974, (in Russian).
2. Н. Кристофидес Теория графов Moscow, 1978, (in Russian).
3. Л. Форд, Д. Фалкерсон Потоки в сетях Moscow, 1966, (in Russian).

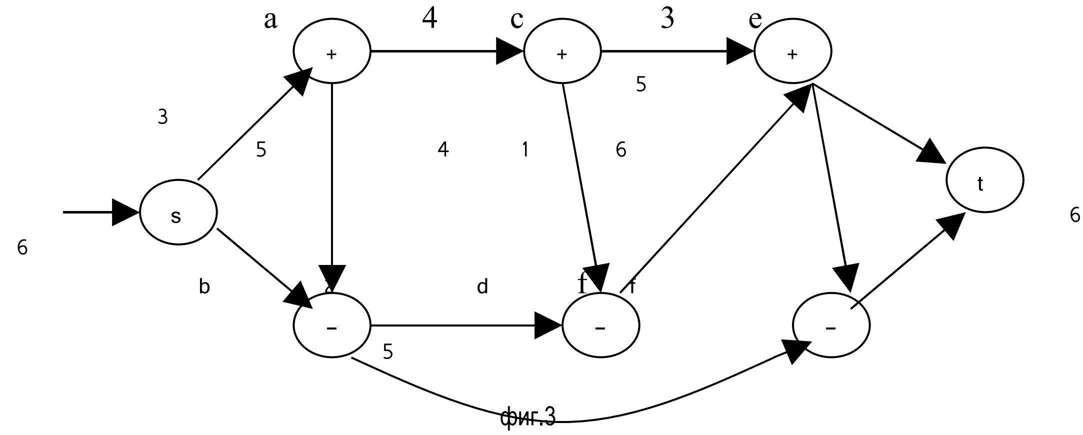
V. Приложения



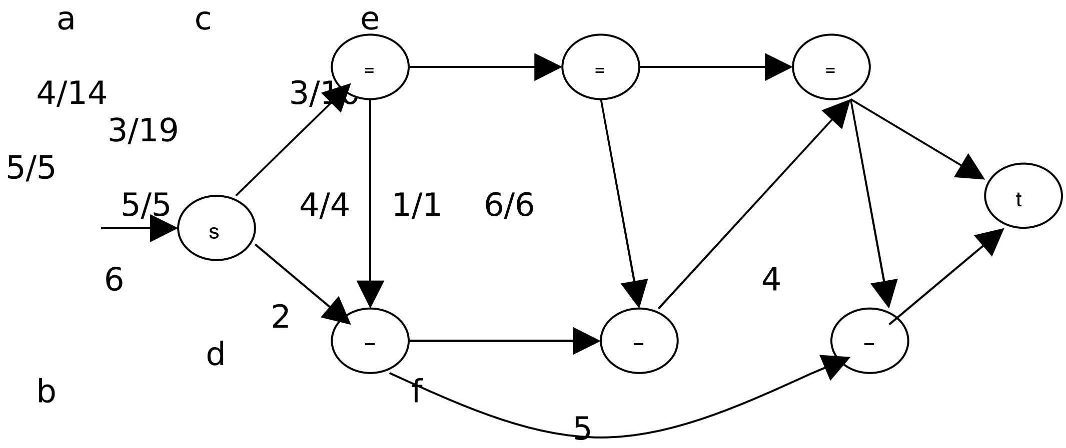
фиг.1



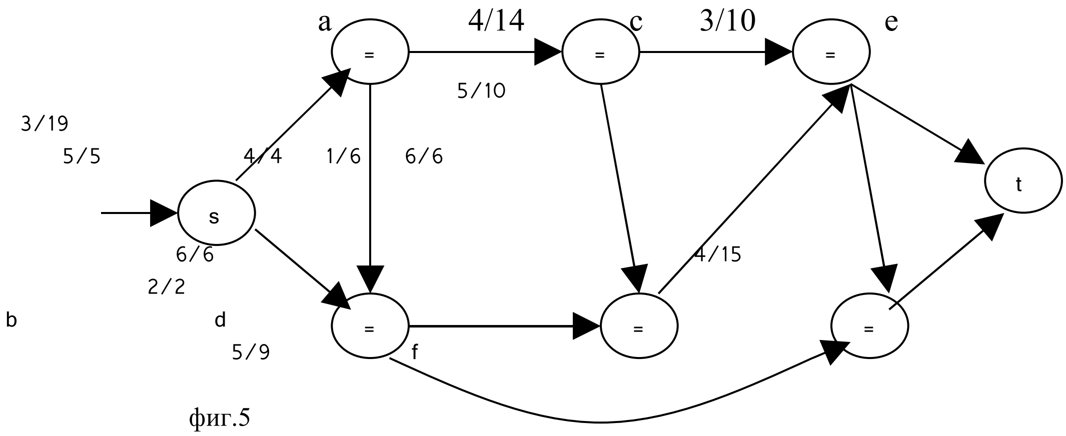
фиг.2



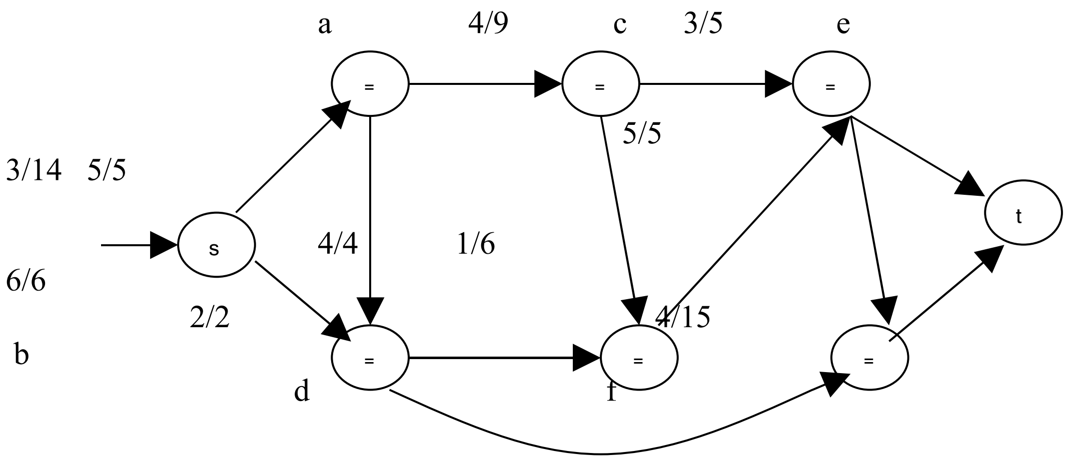
фиг.3



фиг.4



фиг.5



фиг.6