

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 4

ТЕОРИЯ НА ГРАФИТЕ

1. Основна теория

Един от най-удобните начини за представяне на телекомуникационите мрежи и системи е чрез графи. Всеки граф е изграден от възли (точки, върхове) и ребра, които ги съединяват и могат да бъдат насочени (еднопосочни) или ненаочени (двупосочни). Ще представим накратко определенията на някои основни понятия от теорията на графите.

Два възела, съединени с ребро се наричат *съседни*. Броят на ребрата, инцидентни на даден връх се нарича *ранг на този връх*. Вързел с ранг 1 е *краен възел (интерфейсна точка)* и през него не могат да преминават транзитни пътища.

От възел X до възел Y съществува връзка, ако е възможно да се предаде информация директно или чрез посреднически възли. За да се осъществи връзката е необходим *път* - крайната последователност от ребра, започвайки от възел X и завършвайки във възел Y, при което не се преминава два пъти през един и същи възел. Пътят, избран за доставка на съобщение между два възела се нарича *маршрут*. Броят на ребрата, образувачи даден път се нарича *ранг на пътя*. Минималният ранг на път е 1, а максималния – броя на възлите минус единица.

Свързан граф е този, в който кои да е два възела са свързани с поне едн път. Графът е *h-свързан*, ако кои да е два възела от него са свързани с най-малко *h* независими пътя.

Сечение е минималната съвкупност от ребра, които трябва да се извадят от графа, за да се наруши неговата свързаност. Записва се като множество на влизащите в него ребра. Сечение по отношение на два възела е съвкупност от ребра, при отстраняването на които двата възела ще се окажат в различни подграфи. Броят на ребрата, образивачи дадено сечение се нарича *ранг на сечението*. Всички сечения, нарушаващи свързаността между два възела образуват *множество на сеченията* за тези възли.

Обисновено, освен с наименованието си (булев символ), всяко ребро се характеризира и с пропускателната си способност, която често се задава чрез скоростта на предаване на информация в kb/s. *Пропускателна способност на сечение* се нарича сумата на пропускателната способност на отделните ребра, влизащи в него. *Пропускателната способност на път* от възел X до възел Y е равна на пропускателната способност на сечението, което е с най-малка пропускателната способност от всички сечения развалящи връзката от X към Y.

В настоящото упражнение ще определяме и изчисляваме следните параметри при зададени начален и краен възел:

- 1) Всички пътища.
- 2) Оптимален път.
- 3) Сечения.
- 4) Оптимално сечение и максимална пропускателна способност.

Методите, които се използват за определянето или изчисляването им най-общо се разделят в две големи групи – *ресурсоемки* и *оптимизиционни методи*. Първата група методи са подходящи за марки по размер графи (до 8-10 възела), докато втората се препоръчва за големи графи (до 100 възела), поради по-малките изисквания към паметта и по-голямото бързодействие. Изполваните изчислителни методи са подробно обяснени в лекционния материал на дисциплината “Моделиране на телекомуникационни процеси и системи” (МТКПС) от магистърската степен на обучение. Тук ще ги разгледаме накратко.

1.1. Методи за определяне на всички пътища между зададени два възела от граф

В програмата са използванит два метода – рекурсивен метод чрез изчисляване на поддетерминантата (ресурсоемък метод) и метод чрез обхождане в широчина (оптимизиционен метод).

За структурния анализ на телекомуникационните мрежи (намиране на пътища сечения и техните характеристики), представени чрез графи е целесъобразно да се използват структурни матрици, както и някои основни булеви операции с тях. Всеки път (сечение) се представя като конюнкция на ребрата, образуващи този път (сечение).

❖ рекурсивен метод чрез изчисляване на поддетерминантата

Матрицата на свързаност на графа се състои от две матрици, които са в пълно съответствие една спряма друга и са с размерност $n \times n$, където n е броя на възлите. Първата матрица е структурна и чрез нея задават наименованията на ребрата между възлите, т.е. описан е начинът на свързване. Втората матрица съдържа тегловните коефициенти на ребрата, напр. числените капацитети на свързващите линии в kb/s. Рекурсивният метод чрез изчисляване на поддетерминантата изисква от *матрицата на свързаност* да се намери поддетерминантата, която не включва стълба на началния възел и редицата на крайния възел. Определят се всички комбинации от маршрути, на брой $(n-1)!$, и след премахване на циклите се получават всички възможни пътища. Броят им в най-тежкия случай, когато всички възли са взаимосвързани, е експоненциално зависим от броя на възлите. Предимствата на този метод са простия математически модел и върможността за работа без помощта на автоматизирани изчислителни машини. Той е принадлежи на групата на ресурсоемките (неоптимизиран) методи.

❖ метод чрез обхождане в ширина/дълбочина

Този метод решава задачата за намиране на всички пътища между два възела чрез алгоритмичната схема обхождане на граф. Това процедура, при която систематично, по определени правила "се посещават" всички върхове на графа. Съществуват два вида обхождане, които могат да се използват като алгоритмични схеми.

Обхождане в ширина - съществено за тази алгоритмична схема е понятието *ниво на обхождане*, което представлява подмножество на множеството от върховете на графа, което в резултат на обхождането в ширина, се разбива на нива. *Дърво* се нарича краен свързан ацикличен граф имаш поне два върха. Ако даден граф е дърво, то той има един *скелет (покриващо дърво)*, който съвпада с него. Само свързаните графи притежават скелет. В общия случай скелетът на графа не е определен еднозначно. Под обхождане на връх разбираме присъединяването му към дървото, съставено от по-рано обходени върхове. И така, даден е свързан граф, без всякакви ограничения определен връх се избира за корен на бъдещото покриващо дърво на графа, построяването на което е крайния резултат от алгоритмичната схема обхождане в ширина.

Обхождане в дълбочина – резултатът е отново изграждането на покриващото дърво, но то е по-високо и по-малко разклонено от полученото чрез обхождане в ширина.

И двата подхода строят коренови дървета но имат принципни различия. Обхождането в ширина е неудобно поради това, че е необходимо да се помнят всички обходени върхове (или поне върховете от последното обходено ниво), за да може да се построи следващото ниво. За обхождането в дълбочина са достатъчни само върховете, разположени на пътя от началния до текущия връх. Затова при ограничена компютърна памет за предпочитане са алгоритмите, построени по схемата в дълбочина. От друга страна, ако обхождането се прави с цел да се намери връх със зададени свойства и той се окаже на ниво с по-малък номер, тогава алгоритъмът в ширина ще го намери много по-бързо. Търсенето в дълбочина ще работи непредсказуемо дълго, защото може да се окаже, че връхът е измежду последните обходени (дори и да е на сравнително ниско ниво). При този вид задачи за предпочитане е

схемата в ширина. Като компромис между двете схеми могат да се разработят *хибридни обхождания*, съчетаващи добрите страни на едната и другата схема, неутрализиращи, доколкото е възможно недостатъците им. Например, бързо спускане в дълбочина и след това обхождане в ширина на малкото оставащи върхове от "дъното". Използваната в настоящото упражнение програмна реализация в частта си оптимизирани методи използва обхождане в дълбочина за намиране на всички пътища от началната до крайната точка.

❖ методът на Дейкстра

Алгоритъмът на Дейкстра намира оптималните пътища от началото до коя да е точка от графа. Определя се най-ниската цена за превоз от възел 1 до възел i за всички $i = 1, \dots, n$, като необходимото време за работа на алгоритъма (т.е. броят аритметични операции) е от порядъка на n^3 .

Същността на алгоритъма се състои в използването на допълнителна булева променлива за всеки възел, чрез която в процеса на работата една част от възлите се паркират като "обходени", докато такива не станат всичките възли. За всеки отбелязан възел i се записва най-евтината цена $v[i]$ за маршрут, съединяващ $1 \rightarrow i$. Гарантирано е, че този минимум се достига за маршрут, минаващ само през отбелязани възли. За всеки неотбелязан възел i се записва най-евтината цена $v[i]$, получена измежду всички маршрути $1 \rightarrow i$, при които като междинни възли се използват само отбелязани. Множеството на отбелязаните възли се разширява постепенно, благодарение на следното: Ако измежду неотбелязаните възли е i , чието число $v[i]$ е минимално, то това число е равно и на най-евтината цена за преминаването $1 \rightarrow i$. За да докажем, че това е вярно, допускаме, че съществува по-къс (т.е. с по-ниска цена) път от 1 до i . Да разгледаме първия неотбелязан възел i' върху този път. Ясно е, че цената на маршрута $1 \rightarrow i'$ е по-голяма от цената на $1 \rightarrow i$, и следователно (понеже всички цени са неотрицателни) още по-голяма е цената на прехода $1 \rightarrow i' \rightarrow i$. Противоречие.

Алгоритъмът на Дейкстра е по-бавен от директно обхождане на дървото, но има своите предимства, като например намалява зависимостта на методите за намиране на максимално сечение от големината на капацитетите.

1.2. Определяне на сечения и пропускателна способност

❖ метод с булево инвертиране и оптимизиране на резултата

Това е метод за определяне на всички възможни сечения и избиране на най-малкото от тях чрез структурни матрици. От матрицата на свързаност се намират всички възможни пътища и цикли с максимална дължина $n-1$, където n е броят на възлите. Съставя се детерминантата на матрицата след при задраскване на стълба и реда съответно за източника, приемника на информация и ако се изчисли параметрично и булево представлява всички сечения до степен n , където n е броят на точките в мрежата, при това включвайки цикли и повторения. След елиминирането на циклите с просто булево инвертиране на знаците в полученият резултат и минимизиране се получават всички възможни сечения. От тях се избира минималното. Съществено предимство на метода е, че могат да се изброят всички минимални сечения, а недостатък - резултатът само от първата стъпка на изчислителния процес е $(n-1)!$.

❖ алгоритъм на Форд-Фулкерсон

Има още няколко метода за определяне на максимална пропускателна способност, но алгоритъма на Форд-Фулкерсон е добър с по-голямото си бързодействие и значително по-малката си сложност. Принадлежи към групата на оптимизиционните методи и е създаден през 1956 г., като негови модификации са били едни от най-полезните до около 1996 г. Други възможни решения са алгоритъмът на Едмон - Карп, който премахва зависимостта на алгоритъма на Форд-Фулкерсон от големината на капацитетите, алгоритмите на Голдбърг, Онага и др. Последните двама учени са наши съвременници, поради което и алгоритмите им

са значително по-бързи и се препоръчват за изследване на големи мрежи. На практика за особено големи мрежи няма алгоритъм, който да може да изведе максималната пропускателна способност за приемливо количество време.

Основната идея на алгоритъма на Форд-Фулкерсон е да се запаметява състоянието на системата при избора на всеки път, тоест използваните и оставащи капацитети по всяко ребро от пътя. Ето и неговите стъпки.

1. Намираме произволен път от началната точка до крайната точка, който е положителен. Ако такъв не съществува, значи обхождането е завършило и резултатът е получен.
2. Изчисляваме V - максималния поток по този път, който би бил равен на най-малкия капацитет на коя да е дъга по този път и играе ролята на гърло на бутилка.
3. Изваждаме V от оставащия капацитет в права посока на всички дъги от пътя. Добавяме V към оставащия капацитет на всички дъги в обратна посока от пътя.
4. Връщаме се към точка 1.

При изход сумата от потоците от пътищата, намерени по време на стъпка 1 дава максималния поток между началната и крайната точка. Алгоритъмът се прекратява след намирането на последния път. На последната стъпка от алгоритъма на Форд-Фулкерсон - установяването на липса на път, се получава и минималното сечение (спрямо началото), тъй като знаем кои точки са свързани с началото и кои - не.

Теоремата за максималния поток / минималното сечение ни дава основание да използваме алгоритъма на Форд-Фулкерсон за намиране на числената стойност на минималното сечение. Но важен въпрос е и къде, всъщност, се намира то. Ако се окаже, че трафикът необходим между два възела на мрежата е по-голям от капацитета на линиите, които ги свързват, намирането ще покаже кои са линиите, които го ограничават (възможно е да се окаже, че поради неизползвани резерви е необходимо да се разшири само една линия) и съответно ще даде отговор на въпроса кои линии трябва да се оптимизират.

2. Програма ТКМ - flexgrid.

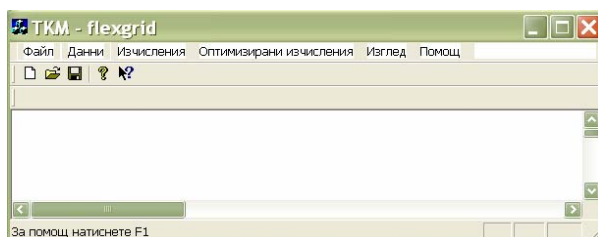
2.1. Цел, резултат и начин на работа

Програма ТКМ - flexgrid е създадена с цел обучение и може да се ползва от преподавателите и студентите на ФКТТ при ТУ - София. Тя има следните възможности - намиране на всички пътища между два върха от граф, сечението им в случай, че са зададени теглови коефициенти, намиране на оптимален път и максимална пропускателна способност.

Необходимите за програмата входни данни са - брой точки в графа; матрица на свързаност (съседство); теглови коефициенти (за намиране на оптимален път и масимална пропускателна способност) и началната и крайна точки, за които ще се търсят решения.

2.2. Основни инструкции при работа с програмата

Програмата е съобразена с интуитивния подход на потребителските интерфейси на "Windows". Командите се въвеждат чрез меню-панела, който се намира в горната част на екрана или панела с икони, намиращ се под него. Меню - панела е изграден от шест главни части (фиг.4.1). При натискане с основния бутон на мишката върху някой от тях се появяват допълнителни възможности.



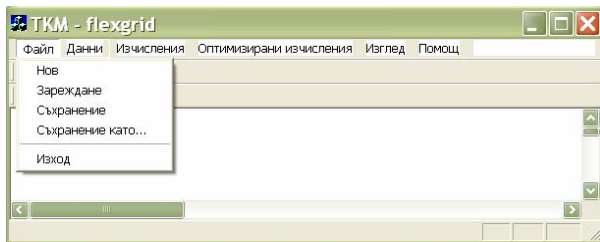
фиг.4.1

Ето кратко описание на шестте опции от главното меню.

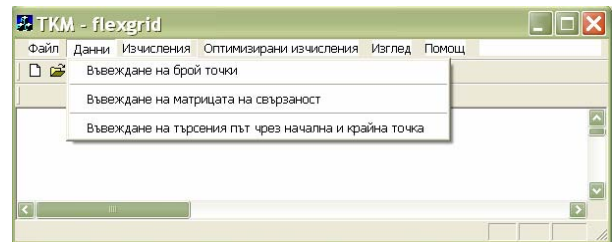
Файл - Основно меню за изход от програмата, съхраняване на данни и т.н. (фиг.4.2)

- ❖ **Нов** – служи за създаване на нов документ – тоест започване на работа отново.
- ❖ **Зареждане** – въвеждане на вече записани файлове.
- ❖ **Съхранение** – запазва текущото състояние на системата, включително резултатите от изчисленията във файловата система.
- ❖ **Съхранение като** – запазва текущото състояние на системата, включително резултатите от изчисленията във файловата система с ново име. Полезно при промяна на документ, който желателно е също да се запази.
- ❖ **Изход** – изход от програмата.

Съхраняването и използването на запазени данни е особено ценно в случай на работа с по-сложни телекомуникационни мрежи. Например, при работа с 10 точки, съответната матрица на свързаност е 10x10 и изисква въвеждането на 90 имена на връзки (10 остават в главния диагонал и са запълнени по начало), както и 90 стойности за пропускателната способност на всяка връзка. Този процес не е особено приятен и е желателно да се избягва. В случай че е възможно, е препоръчително модифицирането на подобни, вече въведени системи. С тази цел е дадена възможност за запаметяване на въведена мрежа във файловата система, както и възможност за нейното прочитане от там в случай на нужда.



фиг.4.2

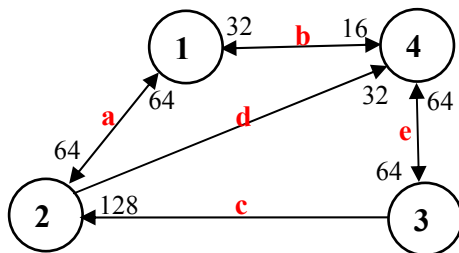


фиг.4.3

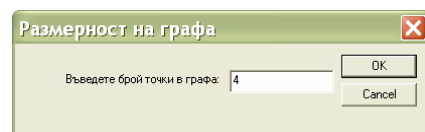
Данни - Основно меню за задаване на необходимите данни за изчисление. Необходимо е да бъдат зададени начални условия за изследване на топологичните параметри на системата. Всички подменюта от тази част са документирани в отделни глави към помощната част на програмата (фиг.4.3).

- ❖ **Въвеждане на брой точки** - отваря се диалогов прозорец с едно текстово поле и два бутона. Въвеждането на броя на точките става в текстовото поле, с горния бутон потвърждаваме, а с долния се отказваме. При натискането на кой да е бутон, ако информацията е зададена коректно, се връщаме в основното положение на програмата. Броят на точките за текущата версия на програмата е цяло число между 1 и 100;

Ще продължим с обясненията на основните инструкции на програмата, разглеждайки малката конкретна мрежа от фиг. 4.4. Броят на точките ѝ е 4 (фиг.4.5).



фиг.4.4



фиг.4.5

- ❖ **Въвеждане на матрицата на свързаност** - след като се зададе броят точки, трябва да се зададе и начина на свързване, както и евентуално тегловете коефициенти на телекомуникационните капацитети на свързващите линии в kb/s. Второто е необходимо

единствено, ако се интересуваме от числени резултати, т. е. за намирането на всички пътища това не е необходимо.

Въвеждането става с диалогов прозорец, който се състои от две таблици с размерност $n \times n$, където n е броя на точките (фиг.4.6). Чрез горната таблица се задават наименованията на ребрата между точките. **0** означава, че ребро не съществува. Допустими са означения на ребра само с главни и малки латински букви, които програмата различава като различни символи. От точка с по-малък номер към точка с по-голям се използват малките латински букви, а в обратна посока се приемат съответните главни латински букви. Например, за брежата от фиг.4.4, реброто между точка **1** и точка **4** е “**b**” и в първи ред, четвърти стълб на първата таблица въвеждаме “**b**”. Между точка **4** и точка **1** е “**B**” и в четвърти ред, първи стълб на същата таблица въвеждаме “**B**”.

В долната таблица се въвеждат числените капацитети. Съпоставят се буквените означения от първата таблица с цифрените стойности от втората. Двете таблици представляват матрицата на свързаност на графа. Капацитетът задаваме в долната таблица, като между точка **1** и точка **4** той е 16 kb/s, докато в обратна посока – от точка **4** към точка **1** е 16 kb/s (фиг.4.6).

Отново за потвърждение се натиска горния бутон “OK”, а за отказ – “Cancel”

Въведете матрицата на свързаност:

	1	2	3	4
1		1 a		0 b
2	A		1	0 d
3		0 c		1 e
4	B		0 E	

Въведете пропускливостите на пътищата тук: (kb/s)

	1	2	3	4
1	Inf	64	0	16
2		64	Inf	0
3		0	128	Inf
4		32	0	64

фиг. 4.6

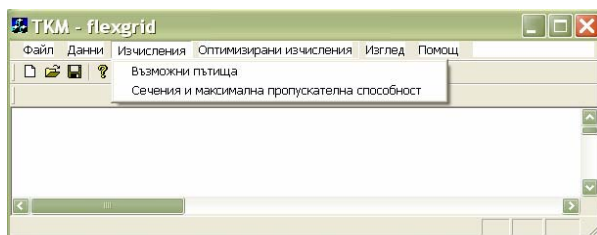
- ❖ **Въвеждане на търсения път чрез начална и крайна точка** - Последната стъпка преди изчисленията е задаване на желанния генератор на трафик и желанния консуматор на трафик - начална и крайна точка. В зависимост от избора им, както пътищата, така и максималната пропускателна способност ще бъдат различни. Отваря се диалогов прозорец с две текстови полета и два бутона (фиг. 4.7). Номерът на началната точка се въвежда в горното текстово поле, а на крайната - в долното. С бутон “OK” потвърждаваме, а с “Cancel” се отказваме от промяната. При натискането на кой да е от тях, ако информацията е зададена коректно, се връщаме в основното положение на програмата. За разглеждания пример избираме начална точка 1 и крайна – 3.

Въведете начална точка: 1

Въведете крайна точка: 3

фиг. 4.7

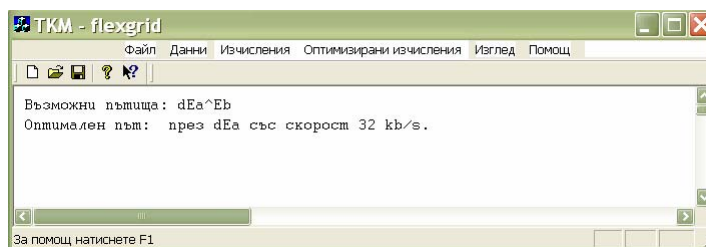
Изчисления - меню за изчисляване на топологичните параметри с ресурсоемки методи. Поради своя характер, изчисленията с тези методи имат недостатъка да са ограничени до 8 точки, което прави това меню е по-скоро демонстративно (фиг.4.8).



фиг.4.8

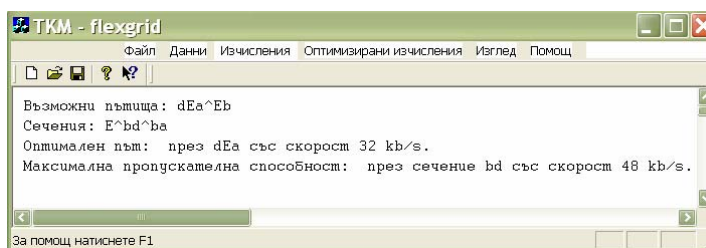
Определят се:

- ❖ **Възможни пътища** – на екрана се отпечатва желаният резултат (фиг.4.9). Използван е методът с изчисление на поддетерминантата. Вижда се, че възможните пътища от точка **1** до точка **3** са два **dEa** и **Eb**. Символът \wedge означава булево “или” (булево “и” не е предвидено като символ в програмата). Подредбата на буквените имена на ребрата не следва точната им последователност за изграждане на пътя. Например, **dEa** означава път със следната последователност: точка **1** – точка **2** (ребро **a**); точка **2** – точка **4** (ребро **d**) и точка **4** – точка **3** (ребро **E**). Или пътят **dEa** може още да се обозначи така 1-(a)-2-(d)-4-(E)-3. Както се вижда още от фиг. 4.9, това е и оптималния път със скорост 32 kb/s.



фиг.4.9

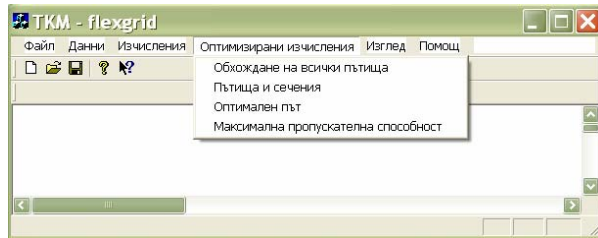
- ❖ **Сечения и максимална пропускателна способност** - изчислява не само възможните пътища и оптималния път, но и сеченията и максималната пропускателна способност. За изчисляване на пътищата отново се използва методът с изчисление на поддетерминантата, а за сеченията - булево инвертиране и оптимизиране на резултата (фиг.4.10). И за двата случая е разумно да се постави ограничение от максимум 8 точки в мрежата с цел изчислителния да не отнема много време.



фиг.4.10

Изчислените три сечения показват, че разрушаването на пътя от точка **1** до точка **3** може да стане по един от следните три начина – като се отстрани реброто **E** (4-3), като се отстранят ребрата **b** и **d** (1-4 и 2-4) или като се отстранят ребрата **b** и **a** (1-4 и 1-2). От трите, сечението **bd** е сумарно с най-ниска скорост ($16+32=48$ kb/s), т.е. то осигурява максимална пропускателна способност.

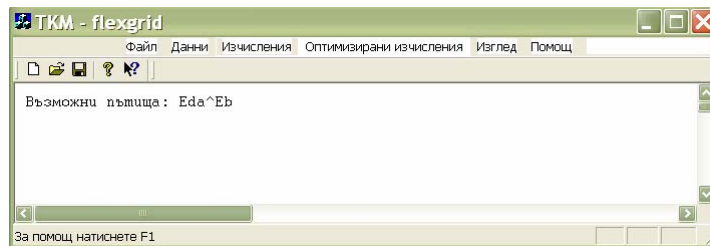
Оптимизирани изчисления - изчисляване на топологичните параметри с нересурсоемки (оптимизирани) методи. Поради своя характер и предимства изчисленията с тези методи са препоръчителни (фиг.4.11).



фиг.4.11

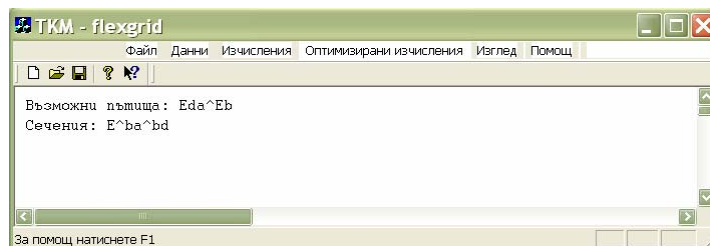
Менюто оптимизирани изчисления се състои от следните подменюта:

- ❖ **Обхождане на всички пътища** - на екрана се отпечатва желаният резултат (фиг.4.12). Използван е метод с обхождане в дълбочина. Въпреки, че се използва различен метод се вижда, че логично резултатите за възможните пътища между точки **1** и **3** съвпадат с тези от фиг.4.9. Единствено последователността на наименованията на ребрата е различна;



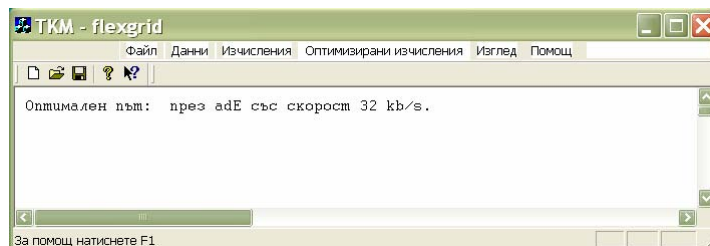
фиг.4.12

- ❖ **Пътища и сечения** – за изчисляване на пътищата е използван отново методът с обхождане в дълбочина, а за сеченията - булево инвертиране и оптимизиране на резултата (фиг.4.13);



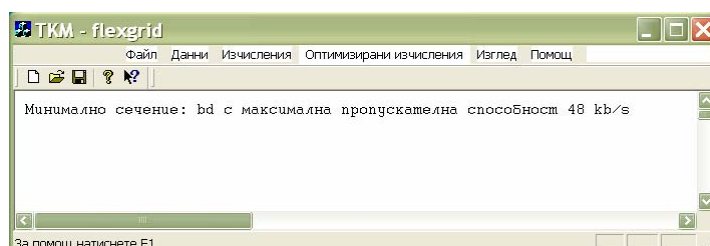
фиг.4.13

- ❖ **Оптимален път** – за определянето му е използван е методът на Дейкстра (фиг.4.14);



фиг.4.14

- ❖ **Максимална пропускателна способност** – за определянето му е използван методът на Форд-Белман (фиг.4.15).



фиг.4.15

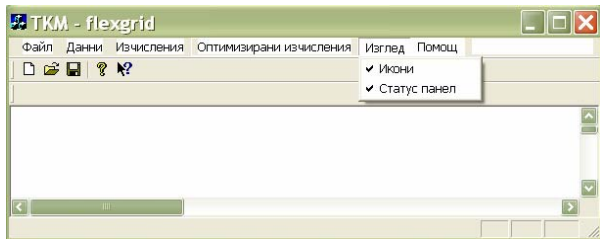
При оптимизираните изчисления единственото ограничение за броя точки е те да са до 100. При силно свързани, големи мрежи големият брой точки може да доведе до значително нарастване на изчислителното време.

Изглед - меню за настройване на визуалните параметри на програмата спрямо желанието на потребителя (фиг.4.16). Съдържа следните подменюта:

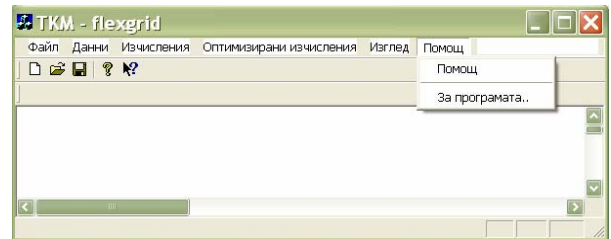
- ❖ **Икони** - служи за скриване, показване на панела с иконите.
- ❖ **Статус панел** - служи за скриване или показване на статус панела, който се намира в долната част на прозореца на програмата.

Помощ - меню с цел помагане работата с програмата, даване на допълнителна информация, опит за олесняване на потребителя (фиг.4.17).

- ❖ **Помощ** – помощна система на програмата;
- ❖ **За програмата** - информация за версията на програмата.



фиг.4.16

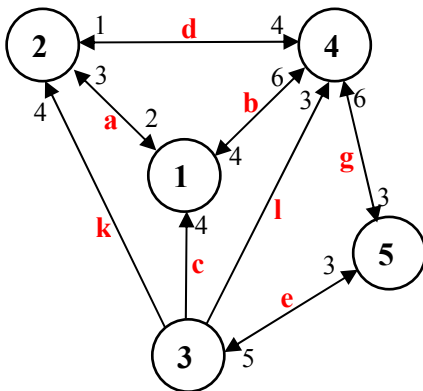


фиг.4.17

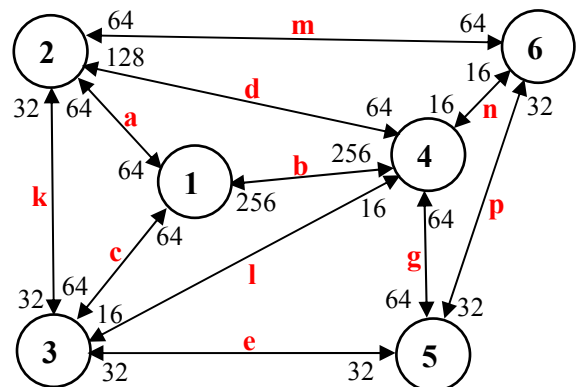
3. Задачи за изпълнение

Задача 1

Дадени са телекомуникационните мрежи:



фиг. А



фиг. Б

И чрез двата вида методи – ресурсоемките и оптимизиционните, намерете:

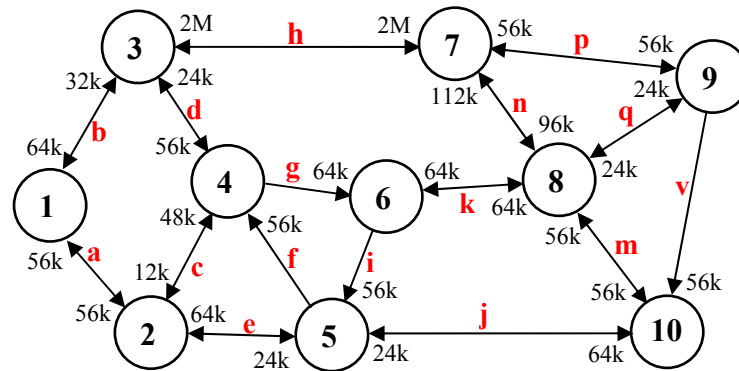
- ❖ всички възможни пътища;
- ❖ оптималния път;
- ❖ сеченията и;
- ❖ максималната пропускателна способност

от точка 2 до точка 5 за мрежата от фиг. А и от точка 3 до точка 6 за тази от фиг. Б.

С малки латински букви са означени ребрата от точка с по-малък номер към точка с по-голям. В обратна посока да се приемат съответните главни букви.

Задача 2

За телекомуникационната мрежа:



определете оптималния път и максималната пропускателна способност между точките:

- ❖ 1 и 10;
- ❖ 2 и 9;
- ❖ 8 и 4.

Означете ребрата с малки латински букви от точка с по-малък номер към точка с по-голям и с главни латински букви в обратна посока.