ГОДИШНИК на Минно-геоложкия университет “Св. Иван Рилски”, Том 49, Св.III, Механизация, електрификация и автоматизация на мините, 2006
ANNUAL of the University of Mining and Geology “St. Ivan Rilski”, Vol. 49, Part III, Mechanization,electrification and automation in mines, 2006

ПРОЕКТИРАНЕ НА КОМПЮТЪРНА МРЕЖА ЗА ИЗЧИСЛЕНИЯ И СЪХРАНЕНИЕ НА ДАННИ С ОПТИМАЛНА ПРОИЗВОДИТЕЛНОСТ ПРИ МИНИМАЛНИ РАЗХОДИ ЗА РЕАЛИЗАЦИЯ

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. При проектирането на компютърни мрежи, включващи възли за съхранение на данни и/или за реализация на изчислителни процеси, се търси възможност за постигане на оптимална производителност по отношение на обема на информацията, която се обменя между възлите и скоростта на обработване на заданията. За целта се предлага абстрактен модел за възстановяване на граф по зададени степени на върховете и търсене на оптимално възстановяване по критерии за оптимална производителност и минимални разходи.

DEVELOPING AN OPTIMAL PERFORMANCE COMPUTER NETWORK FOR DATA STORAGE AND DATA PROCESSING

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

ABSTRACT. In the process of developing computer networks, containing nodes for data storage and/or data processing, opportunities are sought to obtain optimal performance in respect to the amount of information exchanged between nodes and the tasks processing rate. An abstract model is proposed for this purpose describing the reconstruction of graph on given degrees of vertices and seeking the optimal reconstruction upon optimal performance criteria and minimal cost..

Въведение

При проектирането на компютърни мрежи, включващи възли за съхранение на данни и/или за реализация на изчислителни процеси, се търси възможност за постигане на оптимална производителност по отношение на обема на информацията, която се обменя между възлите и скоростта на обработване на заданията.Паралелно с това се цели да бъдат минимизирани разходите по реализацията на мрежата.

Математичен модел

Нека имаме n на брой потокови източника (сървъри,възли за съхранение на данни). На тях съответстват върховете на един граф. На всеки два върха i и j е съпоставено неотрицателно число c_{ij}, което се разглежда като разход за свързването на i-ти и j-ти връх(стойност на връзката). За всеки връх е зададена степен, съответстваща на капацитетните му възможности. Трябва да се генерира мрежа с минимални сумарни разходи. Ако степента на върха v_i е p_i (i=1,2,.....,n), то тази задача може да се формулира като задача на целочисленото математическо програмиране по следния начин:

$x_{ij}=0/1$ е двоична променлива, която приема стойност 1, ако i-ти и j-ти връх се свържат и 0, ако не се свържат. Очевидно $x_{ij}=x_{ji}$.

Общите сумарни разходи са

$$R=\sum \sum x_{ij} c_{ij} \qquad (1)$$

Ограниченията за степените се дават с:

$$\sum x_{ij} = p_i \qquad (2)$$

Търси се min R при ограничения (2).

Този дискретен модел има трудна реализация. Известните методи (на Гомори, на отсичането) изискват много добри компютърни програми, каквито за голямо съжаление не са широко разпространени. Освен това оптималното решение е възможно да не е свързан граф, което е важно за практическите приложения на този модел. Затова ще дадем един алгоритъм,който има предимството,че е лесно реализуем.

Така ако е известна матрицата на свързаност, или инцидентност, не е проблем генерирането на графичния модел на мрежата. Не за всички известни инварианти това е лека задача. При това решението може да не е единствено.

Ще се спрем по-подробно на инвариантата вектор от степените на един граф, която има и конкретно практическо приложение. И така искаме да възстановим един граф G(V,B) с n върха и m дъги, за който е известен вектора от степените на n-те върха s $(p_1,p_2,.......,p_n)$. Преведено на езика на мрежовия анализ, дадени са n елемента, за които трябва да се проектира такава система от връзки, която да удовлетворява изискванията за p_i за всеки елемент(връх). Например, нека имаме n сървъри и искаме да ги обединим в система,така че в момент на претоварване на мрежата, да се използват запасите на всеки от тях. Освен това за всеки сървър, в зависимост от неговите потенциални възможности е поставено и условието с колко на брой сървъри може да бъде свързана. На фиг.1 са дадени два графа, които са възстановени от вектора s(2,2,2,2,2,2).

Този пример показва, че решението не е единствено, двата графа не са изоморфни. Нещо повече, първият дори не е свързан. Все пак интерес представлява алгоритъма за генерирането на граф, с предварително зададен вектор на степените на върховете. Ще опишем един такъв алгоритъм, който е даден в [1], след което използвайки този алгоритъм ще обобщим поставената задача, като въведем и допълнителен критерий за оптималност.

Нека е зададен вектора от степени s($p_1,p_2,.......,p_n$) на върховете. Прието е да се казва, че редицата ($p_1,p_2,.......,p_n$) от неотрицателни цели числа е графическа, ако съществува граф с върхове $v_1,v_2,.....,v_n$, такъв че върха v_i има степен p_i. Не е трудно да се прецени, че сумата от степените p_i трябва да е четно число.

Номерираме отново върховете на графа в съответствие с наредбата:

$$p_1 \geq p_2 \geq .... \geq p_n$$

Да вземем върха v_i със степен p_i Да ***изчерпим*** p_i означава да свържем върха v_i с върховете $v_1,v_2,......,v_{pi}$, ако $p_i<i$, или с $v_1,v_2,\ldots v_{i-1},v_{i+1},..,v_{pi+1}$,ако $pi \geq I$. Редицата (p_1-1,p_2-1,......,p_{pi}-1,p_{pi+1},...,p_{i-1} ,0,p_{i+1} ,....,p_n) се нарича **остатъчна** редица. След тези определения може да формулираме и алгоритъма за реализация на редицата ($p_1,p_2,........,p_n$) , $p_1 \geq p_2 \geq ..... \geq p_n$

1. избираме произволно $p_k \neq 0$. Изчерпваме p_k в дефинирания смисъл;

2. определяме остатъчната редица и подреждаме отново върховете, така че остатъчната редица да стане намаляща. Повтаряме този процес, докато стигнем до една от следните две ситуации:

- всички остатъчни степени са 0. В този случай графа е построен;
- една от остатъчните степени е отрицателна. Следователно редицата s не е графическа.

Ако в описания алгоритъм изчерпваме на всяка стъпка най-малката ненулева остатъчна степен, полученият граф ще бъде свързан.

Ясно е, че решението на разглежданата задача не е еднозначно. Тогава можем да въведем изискването за оптималност на някой от вариантите на решение. Ще се спрем именно на този проблем и неговото практическо приложение.

Нека са дадени вектора от степените на дадени n върха s=($p_1,p_2,...,p_n$) и симетричната матрица { c_{ij} } от разходите за свързване на i-тия и j-тия връх. Сега спазваме следната последователност:

1. като начало се построява свързан граф G_0 с дадения вектор на степените, използувайки дадения по-горе алгоритъм;
2. изчисляват се разходите за реализация на тази мрежа $R_{0;}$
3. дъгите на графа G_0 се оцветяват.

G е пълния граф, образуван от n върха. След оцветяването на G_0, част от дъгите на G са оцветени, а останалите не са. По нататък ще използуваме една теорема. Преди нейното изложение ще дадем една дефиниция.

Алтернативен контур в G ще наричаме такъв контур в който оцветяването на дъгите, участвуващи в него се сменя алтернативно. На фиг.2 е показан такъв контур.

Теорема: Ако в алтернативен контур на G сменим оцветените дъги с неоцветени и обратно, новия граф от оцветени дъги е реализация на вектора s.

Доказателство Ако вземем произволен връх от алтернативен контур, той е инцидентен с една оцветена и една неоцветена дъга. Промяната на оцветеността на тези две дъги не води до промяна на оцветената степен на този връх.

Ако в един алтернативен контур, сумата от разходите на оцветените дъги е по- голяма от сумата от разходите на неоцветените дъги, този контур ще наричаме **алтернативно намаляващ**.

4. в G търсим алтернативно намаляващ контур. Ако намерим такъв, променяме оцветеността на дъгите в него и получаваме нова реализация на мрежата с по-малки разходи. Същевременно следим за свързаността на новополучения граф.

Точка 4 се повтаря, докато в G не може да се открие повече алтернативно намаляващ контур.

Да разгледаме един пример.

	v_1	v_2	v_3	v_4	v_5	v_6
v_1		16	8	12	10	9
v_2	16		14	13	8	15
v_3	8	14		11	9	7
v_4	12	13	11		10	6
v_5	10	8	9	10		12
v_6	9	15	7	6	12	

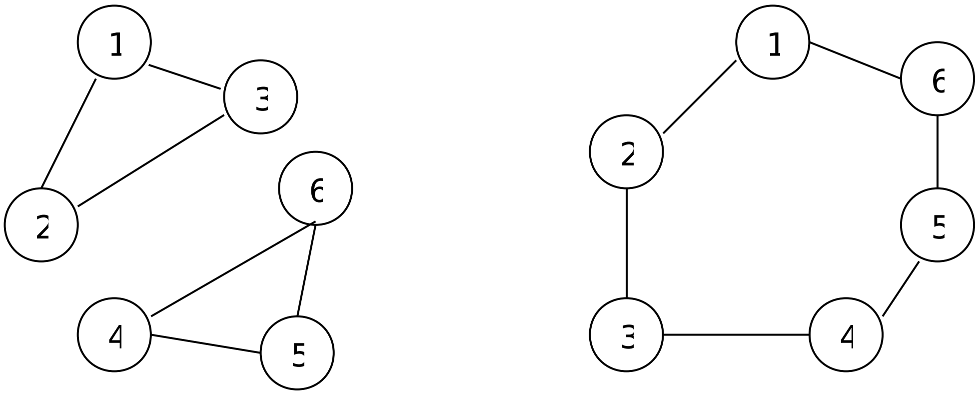
s=(4,3,3,2,2,2)

Генерираме един от графите, по зададения вектор от степени.На фиг.3 се вижда графическата реализация на този граф.

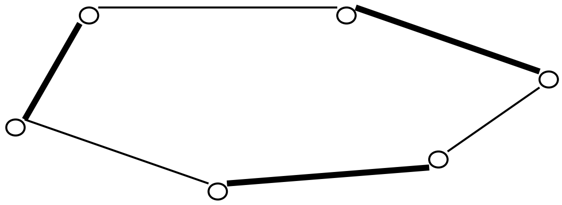
Литература

Р.Басакер, Т.Саати Конечыіе графы и сети, Moscow, 1974, (in Russian).

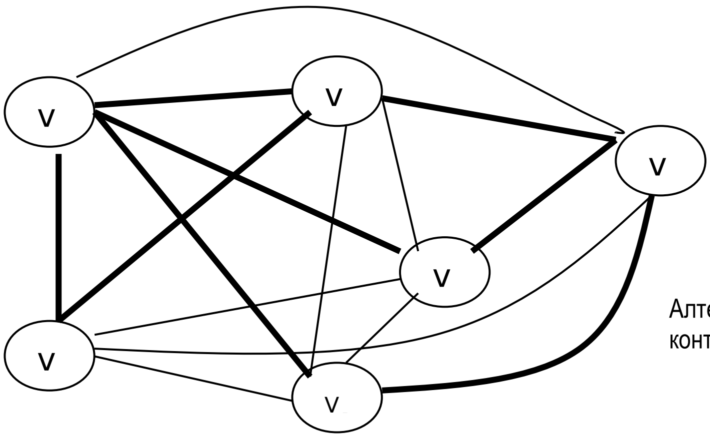
ПРИЛОЖЕНИЕ



Фиг.1

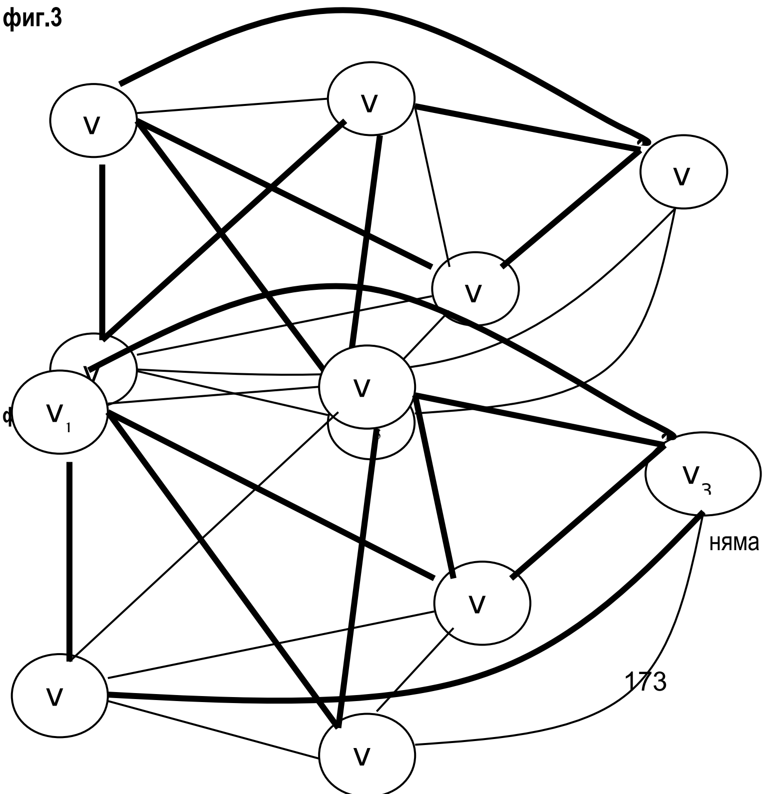


Фиг.2



R_0=96
Алтернативно намаляващ
контур:v_2-v_1-v_3-v_5-v_2

фиг.3



R_0=87
Алт. намаляващ
контур:v_2-v_6-v_5-v_4-v_2

R_0=81
няма алт.намаляващ
контур.

фиг.5

Следователно това е оптимално решение. Намирането на алтернативно намаляващ контур може да стане например с помощта на алгоритъма **ход назад**.

Препоръчана за публикуване от
Катедра "Информатика", МЕМФ