

## ИНДИКАЦИЯ ЦИРКУЛЯЦИОННЫХ КОНТУРОВ В ОРИЕНТИРОВАННОМ ГРАФЕ СЕТЕВОГО ОБЪЕКТА

Назаренко В.И.  
Кафедра ЭВМ ДонГТУ  
vn@cs.dgtu.donetsk.ua

### *Abstract*

*Nazarenko V.I. Indication of circulating loops in an oriented graph of a network object. The problem of automatic search for circulating loops in an oriented graph of a network object is considered. The presence of such loops is in general a sign of dangerous situation in the motion of material flows in network branches.*

При расчете и анализе потокораспределения в сетевом объекте последний изображается как ориентированный граф, используемый в дальнейшем для автоматического формирования узловых и контурных уравнений [1]. Вид компонент таких уравнений зависит от физических свойств материальных потоков, циркулирующих в сети. Примерами сетевых объектов являются системы энергоснабжения, шахтные вентиляционные сети, газораспределительные сети и др.

Граф сетевого объекта является ориентированным, связным, но не сильно связным. Последнее означает, что среди вершин графа можно указать миноранты и мажоранты, т.е. вершины, для которых отсутствуют соответственно входящие или исходящие ветви.

Предметом рассмотрения является индикация возможных циркуляционных контуров, т.е. таких контуров, в которых материальные потоки одинаково ориентированы по отношению к направлению обхода контура. Такие контуры возникают в сети при наличии локальных источников энергии (например, при пожаре в шахтной вентиляционной сети) и являются, как правило, признаком аварийной ситуации. В связи с этим автоматизация поиска циркуляционных контуров в сетевом объекте, имеющем в своем составе несколько сотен ветвей, является важной практической задачей.

В качестве примера на рис.1 представлена упрощенная схема сетевого объекта. На этой схеме 1, 2 и 3,4 - соответственно миноранты и мажоранты, E1 и E2 - основные источники энергии, подчеркнутые числа - номера вершин, остальные числа - номера ветвей. Циркуляционными здесь являются контуры 7 - 26 - 43 - 25, 8 - 21 - 51 - 25 и 24 - 47 - 48.

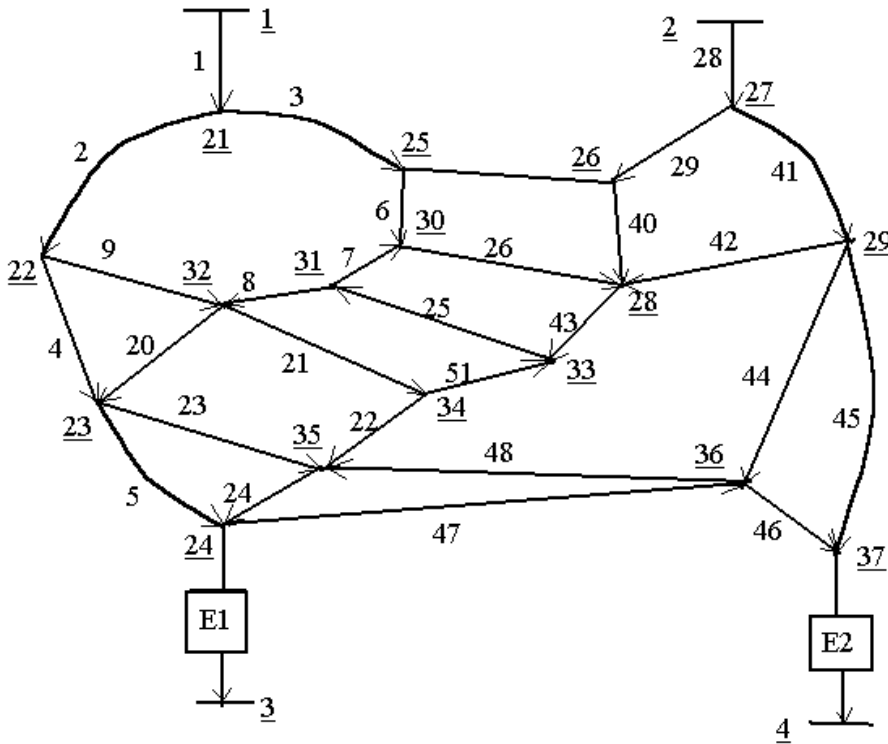


Рисунок 1 - Граф сетевого объекта.

Будем обозначать граф сетевого объекта [2]

$$G = (X, V),$$

где  $X = \{ x_1, x_2, \dots, x_n \}$  - множество вершин графа,  $|X| = n$ ;

$V = \{ v_1, v_2, \dots, v_m \}$  - множество его ветвей,  $|V| = m$ .

При нормальном функционировании объекта граф  $G = (X, V)$  не должен содержать ни одного циркуляционного контура. Если же в графе имеются такие контуры, то в их образовании принимает участие ограниченное подмножество ветвей исходного графа. В связи с этим алгоритм поиска циркуляционных контуров целесообразно разделить на два этапа:

- а) формирование циркуляционного подграфа;
- б) поиск циркуляционных контуров в подграфе.

Циркуляционным подграфом будем называть такой подграф, который образуется из исходного графа путем удаления тех ветвей и вершин, которые наверняка не принимают участия в формировании циркуляционных контуров.

Обозначим циркуляционный подграф  $D = (Y, U)$ , где  $Y \subset X$ ,  $U \subset V$ .

Удаленные ветви и вершины образуют соответственно множества

$$Z = X \setminus Y, \quad W = V \setminus U.$$

Очевидно, что

$$Y \cup Z = X, \quad U \cup W = V.$$

Примем начальные значения  $Y = X$ ,  $U = V$ .

Циркуляционный контур не может проходить через миноранту и мажоранту. Поэтому на первом этапе формирования циркуляционного подграфа миноранты и мажоранты исходного графа  $G = (X, V)$  включим в множество  $Z$ , а инцидентные им ветви - в множество  $W$ , исключив соответствующие элементы из множеств  $Y$  и  $U$ .

Будем циклически просматривать вершины  $x_i \in X$ . Ветви, инцидентные вершине  $x_i$ , обозначим  $v_i^+$ ,  $v_i^-$ , где  $v_i^+$  - ветвь, входящая в вершину  $x_i$ ,  $v_i^-$  - ветвь, исходящая из этой вершины. Количество таких ветвей обозначим соответственно  $m_i^+$ ,  $m_i^-$ .

Если  $x_i \in U$  и  $v_{ij}^+ \in W$ ,  $j = 1..m_i^+$ , то циркуляционный контур не проходит через  $x_i$ . Тогда

$$Z = Z \cup \{x_i\}; \quad Y = Y \setminus \{x_i\};$$

$$W = W \cup \{v_{ij}^-\}; \quad U = U \setminus \{v_{ij}^+\}, \quad j = 1..m_i^+.$$

Это означает: если вершина  $x_i$  принадлежит циркуляционному подграфу, а все входящие в нее ветви этому подграфу не принадлежат, то циркуляционный контур не может проходить через вершину  $x_i$  и, следовательно, через исходящие из этой вершины ветви.

Аналогично при

$$x_i \in U \quad \text{и} \quad v_{ij}^- \in W, \quad j = 1..m_i^-$$

выполним

$$Z = Z \cup \{x_i\}; \quad Y = Y \setminus \{x_i\};$$

$$W = W \cup \{v_{ij}^+\}; \quad U = U \setminus \{v_{ij}^-\}, \quad j = 1..m_i^-.$$

Циклический просмотр вершин  $x_i$  производится до тех пор, пока после очередного цикла просмотра не будет отмечено, что количество элементов в множестве  $Z$  не изменилось. Если при этом  $Y = \emptyset$ , то граф  $G = (X, V)$  не содержит циркуляционных контуров, в противном случае их поиск должен осуществляться на подграфе  $D(Y, U)$ .

Поиск циркуляционных контуров в  $D(Y, U)$  будем производить, просматривая в цикле вершины  $y_i \in Y$ . Методика поиска сводится к последовательному построению пути, начинающемуся в вершине  $y_i$ .

При поиске контура будут использованы два стека:  $P$  и  $S$ . В стек  $P$  будем включать ветви, добавляемые в формируемый путь; в стек  $S$  - ветви, отходящие от вершин строящегося пути.

Рассмотрим произвольную вершину  $y_i \in Y$ . Запишем в  $P$  ветвь  $u_j$ , отходящую от  $y_i$ . Если для  $y_i$  имеются другие отходящие ветви, то их номера запишем в стек  $S$ . Теперь перейдем к вершине  $y_j$ , конечной для ветви  $u_j$ , и произведем аналогичную работу. Если после добавления очередной ветви  $u_k$  обнаружится, что ее конечная вершина  $y_k$  совпадает с  $y_i$ , то построение контура закончено. Если на очередном этапе будет отмечен "тупик" формирования (нет исходящих ветвей, не включенных в  $P$ ; обнаружено локальное замыкание  $y_k = y_j$ ;  $y_k \neq y_i$ ), то берется ветвь  $u_s$ , последняя из записанных в стек  $S$ , из  $P$  удаляются ветви, записанные после  $u_s$ , а от ветви  $u_s$  продолжается формирование пути.

Циркуляционный контур не проходит через вершину  $y_i$ , если стек  $S$  исчерпан и ни для одного из формируемых путей не обнаружено замыкание на вершину  $y_i$ .

В рассматриваемой задаче представляет интерес не только индикация циркуляционного контура, т.е. построение списка ветвей, входящих в такой контур, но и определение количественного значения материального потока  $Q_c$ , циркулирующего по этому контуру. Для решения второй подзадачи обратимся к рис.2, на котором изображен циркуляционный подграф сети рис.1. Пунктиром на рис.2 показаны ветви, обрезаемые при выделении данного подграфа, т.е. ветви, определяющие связь подграфа с остальной сетью. При этом ветви 4, 6, 9, 40, 42, 44 - входящие, а ветви 46 и 50 - исходящие по отношению к подграфу.

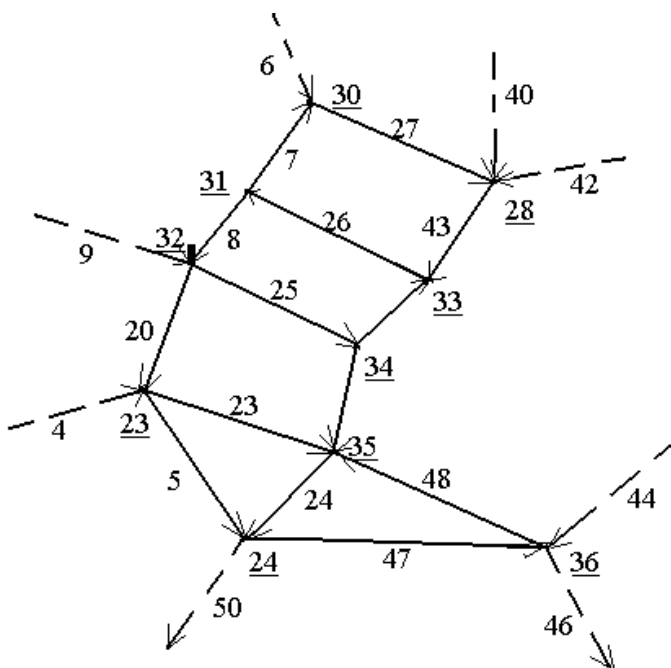


Рисунок 2 - Циркуляционный подграф.

Является очевидным, что вне зависимости от топологии циркуляционного подграфа сумма его входящих потоков равна сумме исходящих. В то же время при одних и тех же значениях потоков во входных и выходных ветвях значения потоков в собственных ветвях подграфа могут быть различными.

Если к потокам в ветвях циркуляционного контура добавить или отнять одни и те же значения, то их баланс в вершинах этого контура не изменится. Вычтем из потоков ветвей, входящих в циркуляционный контур, их минимальное значение. Тогда в одной из ветвей контура установится нулевое значение потока. Это означает, что такой контур уже не является циркуляционным. Следовательно, циркуляционный поток численно равен минимальному потоку в ветвях, образующих циркуляционный контур.

Рассмотренные выше алгоритмы поиска циркуляционных контуров и определения значений циркуляционных потоков реализованы в программной системе, выполняющей расчет и анализ потокораспределения в шахтной вентиляционной сети в нормальном и аварийных режимах ее функционирования.

### *Литература*

1. Цой С., Цхай С.М. *Прикладная теория графов*. - Алма-Ата: Наука, 1971. - 500 с.
2. Липский В. *Комбинаторика для программистов*. - М.: Мир, 1988. - 213 с.