

УДК 517.15

Колесников В.Б., Новоселов В.Г., канд. физ.-мат. наук, проф.

Севастопольский национальный технический университет

Студгородок, г. Севастополь, Украина, 99053

E-mail: kvt@sevgtu.sebastopol.ua

АЛГОРИТМ ПОСТРОЕНИЯ ОСТОВА И СИСТЕМЫ ФУНДАМЕНТАЛЬНЫХ ЦИКЛОВ ГРАФА С ИСКЛЮЧЕНИЕМ ПЕРЕБОРА

Предлагается алгоритм совместного построения остова и системы фундаментальных циклов для каждого компонента связности ненаправленного графа. Благодаря использованию свойств структуры, упорядочению ребер при создании остова и применению произведенного упорядочения при построении цикла в алгоритме исключается перебор, характерный для известных методов.

1. Постановка задачи. Задан неориентированный граф, содержащий n вершин, m ребер и p компонентов связности. Необходимо построить лес и систему фундаментальных циклов [1].

Следуя [1,2], для представления графа воспользуемся так называемой *структурой* – адресным и информационным массивами. $G(x_i)$ – множество вершин, смежных вершине x_i . Массив вершин, представленный последовательностью $G(x_1), G(x_2), \dots, G(x_n)$, называют *информационным*. Массив A , содержащий адреса a_i множеств $G(x_i)$ в информационном массиве, называется *адресным*. Можно обойтись без адресного массива, затратив лишнюю память. Для каждого множества $G(x_i)$ достаточно отвести в информационном массиве количество элементов большее или равное максимальной степени вершин графа (таблица 1).

Использование структуры имеет ряд преимуществ перед другими формами представления графа. В частности, к основным преимуществам структуры можно отнести экономию памяти и повышение быстродействия за счет исключения поиска при нахождении вершин, которые смежны рассматриваемой (в матрице отношения смежности, например, пришлось бы просматривать и нулевые элементы).

2. Введение псевдопорядка ребер в остове. Будем считать, что для одного из компонентов связности графа остов уже построен. Для обеспечения линейной независимости в алгоритме построения системы фундаментальных циклов в каждый из них вводятся $L-1$ принадлежащих остову ребер (где L – длина цикла), и только одно, остову не принадлежащее. Для построения всей системы фундаментальных циклов для каждого, не принадлежащего остову ребра (например, $x_i x_j$) данного компонента связности, необходимо найти остальные $L-1$ ребра остова, составляющих в нем единственный маршрут из вершины x_j в вершину x_i , проходящий через ре-

бра остова. Естественна аналогия между алгоритмом построения цикла и задачей о поиске кратчайшего пути через лабиринт. Для определенности будем считать, что вершина x_j – вход в "лабиринт" (остов), а вершина x_i – выход. Классическое решение подобной задачи состоит в использовании так называемого "алгоритма последовательных испытаний" (например, обход лабиринта по правилу левой или правой руки) – *перебор с возвратом*. Очевидно, что чем больше количество ребер в графе, тем больше вариантов потребуется перебрать и тем глубже будет перебор, а, следовательно, тем меньше будет производительность. Предлагаемое усовершенствование позволяет *вообще* исключить перебор. Его суть заключается в следующем: если бы все ребра остова оказались дугами, направленными по прохождению маршрута в x_i , то путь из x_j в x_i строился бы однозначно на основе последовательности переходов по псевдоориентации дуг остова.

Рассмотрим реализацию данной идеи применительно к процессу совместного построения остова и системы фундаментальных циклов. Для представления остова графа и ориентации, вошедших в него ребер воспользуемся одномерным массивом, i -й элемент которого содержит номер вершины, в которую в остове направлена дуга из вершины x_i . Для корневой вершины номер элемента массива и значение самого элемента совпадают.

Выберем произвольную вершину x_i , еще не вошедшую в предыдущие остовы, и будем считать ее активной вершиной и корнем дерева. Выберем $x_j \in G(x_i)$; если вершина x_j еще не принадлежит текущему остову, то вводим ее в остов с псевдонаправлением ребра $x_j x_i$ к корню, в противном случае строим очередной цикл. Рассмотрим два возможных метода построения цикла.

1) Корневая вершина остова в процессе построения системы фундаментальных циклов не изменяется. В этом случае справедливо следующее утверждение: *для очередного ребра $x_j x_i$, обе вершины которого принадлежат текущему остову, цикл строится посредством последовательного введения ребер остова по их псевдоориентации до тех пор, пока они не пересекутся в одной вершине, быть может, корневой*. Недостатком данного метода является необходимость проверки на пересечение двух маршрутов при добавлении очередного ребра в цикл и, возможно, удаление лишних рассмотренных ребер, не вошедших в цикл по одному из путей.

2) Корневая вершина в процессе построения системы фундаментальных циклов изменяется. При использовании данного метода устраняется проблема проверки маршрутов на наличие точки пересечения, но возникает необходимость в "переориентации" остова графа относительно нового корня. Суть переориентации заключается в следующем: пусть x_i – корень остова, x_j – активная вершина; считаем вершину x_j новым корнем и все дуги

по пути из нового корня к старому корню направляем к новому корню. Следует отметить, что переориентацию остова следует проводить только при смене корня. Справедливо следующее утверждение: для очередного ребра $x_i x_j$ (где x_i — новый корень, x_j — вершина, ему инцидентная), обе вершины которого принадлежат текущему остову, цикл строится, начиная с вершины x_j посредством последовательного введения ребер остова в направлении их ориентации до тех пор, пока не будет достигнута вершина x_i .

3. Формулировка алгоритмов. Сформулируем алгоритмы построения остова и системы фундаментальных циклов графа.

Алгоритм построения остова.

Начало

1. Выбираем из исходного графа очередную вершину, еще не включенную в построенные остовы, включаем ее в новый остов для очередного компонента связности и объявляем текущим корнем в нем; одновременно вводим ее в массив М активных вершин, введенных в текущий остов, но еще не рассмотренных.
2. Выбираем из М очередную вершину x_i . Находим $G(x_i)$ по структуре и для каждой вершины $x_j \in G(x_i)$ выполняем следующие пункты.
 - 2.1. Если вершина x_j принадлежит остову, то строим цикл (алгоритм формирования цикла рассмотрим отдельно).
 - 2.2. Иначе добавляем ребро $x_i x_j$ (с учетом направления), а также вершину x_j в остов; вершину x_j также добавляем в массив М. Следует отметить, что, в связи со спецификой используемой структуры данных, при добавлении очередной вершины в остов соответствующее ребро добавляется неявно.

Конец

Алгоритм построения цикла.

Пусть необходимо построить цикл $x_j \dots x_i - x_j$ для рассматриваемого ребра $x_i x_j$, не принадлежащего остову. Как уже было отмечено выше, остов представляется одномерным массивом, i -й элемент которого содержит номер вершины, в которую псевдонаправлено ребро из x_i .

Начало.

1. Добавляем в цикл вершину x_j .
2. Считая вершину x_i корнем, переориентируем остов графа — все дуги по пути из нового корня к старому корню, направляем к новому корню. Применительно к используемой структуре данных это

будет выглядеть следующим образом: запоминаем содержимое ячейки, адресуемой содержимым i -го элемента массива, затем заносим в нее i . Повторяем этот процесс, используя запомненное значение как номер рассматриваемого элемента, и присваиваем ему номер предыдущего рассматриваемого элемента, до тех пор, пока не обратимся к корневой вершине (номера рассматриваемого и предыдущего элементов массива совпадают).

3. Строим путь от вершины x_j к вершине x_i ; совокупность всех ребер, проходимых при обходе, и будет требуемым циклом.

Конец

4. Пример построения остова и системы циклов. Рассмотрим пример построения остова и системы фундаментальных циклов графа на рисунке 1.

Данному графу соответствует следующая структура (в целях упрощения восприятия в виде таблицы 1):

Пусть в очереди вершин на рассмотрение содержится вершина 1. Ей смежны вершины 4 и 7, и поскольку ни одна из них еще не была рассмотрена, добавляем вершины в очередь и в остов; так как вершина 1 является

Таблица 1 – Представление графа структурой без использования адресного массива

Вершина		Смежные
вершины	1	472683457841365367862457
8		1358235

Рисунок 1 – Исходный граф

корневой по отношению к вершинам 4 и 7, то четвертый и седьмой элементы остова будут содержать единицу.

Следующей в очереди является вершина 4. Вершины 3 и 6, ей смежные, еще не принадлежат остову, поэтому вводим соответствующие вершины и ребра в остов и в очередь заявок. На данный момент остов представлен на рисунке 2:

1	2	3	4	5	6	7	8
1	-	4	1	-	4	1	-

Рисунок 2 – Остов графа

Следующая вершина (седьмая) смежна вершинам 5 и 3. Но вершина 3 уже принадлежит остову, поэтому переориентируем остов и строим цикл, считая вершину 7 новым корнем (рисунок 3). Производим обход остова от вершины 7, получаем цикл 7-3-4-1-7.

Элемент	1	2	3	4	5	6	7	8
До переориентации	1	-	4	1	-	4	1	-
После переориентации	7	-	4	1	-	4	7	-

Рисунок 3 – Остов графа до и после переориентации

Вершину 5 добавляем в остов и в список активных, считая ее направленной к вершине 7. Далее рассматриваем вершину 3 и строим цикл с вершиной 5, считая вершину 3 корнем. В результате получаем цикл 3-5-7-1-4-3. Вершина 8, смежная вершине 5, еще не была рассмотрена, поэтому вводим ее в остов и в список активных. Вершине 6 смежна вершина 2, остову не принадлежащая. Добавляем вершину 2 в остов и в список активных. Вершина 5 принадлежит остову и является его корнем. Строим цикл 6-5-7-1-4-6.

Следующая вершина в очереди – вершина 5. Ей смежна вершина 8, с которой она образует цикл 5-8-3-4-1-7-5. Вершине 8 смежна вершина 2, они образуют цикл 8-2-6-4-3-8.

Все ребра графа рассмотрены, следовательно, остов и система циклов графа построены. В остов вошло $8 - 1 = 7$ ребер, построено $12 - 8 + 1 = 5$ циклов. Проверив полученный результат по теореме Эйлера, делаем вывод, что как остов, так и система фундаментальных циклов были построены правильно.

5. Оценка быстродействия алгоритма. Быстродействие описанного алгоритма можно оценить по формуле:

$$t \propto \sum_{i=1}^v L_i + \sum_{j=1}^n G(x_j).$$

Среднее быстродействие оценивается по уточнённой формуле:

$$t \propto m + vL,$$

где m – число ребер; $v = m - n + p$ – цикломатическое число графа; L – средняя длина цикла. При построении остова в ширину (рассмотрении вершин в порядке их появления в остове и всех инцидентных этим вершинам ребер) средняя длина L циклов уменьшается и, как показывает проведенный статистический эксперимент, равняется 5.

Представленный алгоритм имеет ту же, что и в [1] по порядку оценку быстродействия (более уточненную), но реальное время решения сокращено, так как полностью исключает переборы, не связанные с самими проце-

дурами построения остова и системы фундаментальных циклов. В большинстве известных решений этой задачи, алгоритмы имеют худшие оценки.

Представленный алгоритм построения остова и системы фундаментальных циклов может эффективно использоваться, например, в САПР радиоэлектронных и цифровых схем при построении систем линейно независимых уравнений Кирхгофа токов.

Библиографический список

1. Липский В. Комбинаторика для программистов / В. Липский. — М.: Мир, 1988. — 319 с.
2. Новоселов В.Г. Дискретная математика в задачах и примерах. Ч. 2: Учеб. пособие / В.Г. Новоселов, А.В. Скатков. — Севастополь: Изд-во СевНТУ, 2002. — 126 с.

Поступила в редакцию 23.07.2002 г.