УДК 519.17

ВЕРШИННОЕ ПОКРЫТИЕ ЦИКЛОВ

М.В. Матросов

Нижегородский госуниверситет им. Н.И. Лобачевского

matro_michail@mail.com

Поступила в редакцию 09.04.2013

Показано, что следующая задача может быть решена за линейное от числа вершин время: дан простой граф, найти в нем два вершинно независимых цикла или построить вершинное покрытие циклов, содержащее не более 3 вершин.

Ключевые слова: графы, непересекающиеся циклы, цикловые упаковки, цикловые покрытия.

Введение

Рассмотрим неориентированный граф, не содержащий петель и кратных ребер. Цикловой упаковкой графа называется множество его циклов, попарно не имеющих общих вершин. Наибольшее число циклов в цикловой упаковке графа G будем обозначать через $\nu(G)$. Вершинным цикловым покрытием называется такое множество вершин, что каждый цикл содержит хотя бы одну из этих вершин. Наименьшее число вершин в цикловом покрытии будем обозначать через $\tau(G)$.

Очевидно, что для любого графа выполняется неравенство $\nu(G) \le \tau(G)$. Известны также верхние оценки числа τ в зависимости от ν [1–3]. В частности, если $\nu(G) = 1$, то $\tau(G) \le 3$ [4, 5], и это значение достигается (например, на полном графе K_5). В [4, 5] полностью описано множество графов Ω_2 с $\nu(G) = 1$.

Задача определения по данному графу G и числу k – верно ли, что $\nu(G) \ge k$, – является NP-полной. В работе [1] доказано, что при любом фиксированном k она решается за линейное время. В этой же работе показано, что при отрицательном ответе (т.е. $\nu(G) < k$) можно за линейное время найти вершинное цикловое покрытие, мощность которого ограничена сверху некоторой функцией от k. В частности, при $k = 2$ алгоритм Бодлендера находит не более 9 вершин, покрывающих все циклы. Из упомянутых выше результатов следует, что в этом случае существует покрытие не более чем из 3 вершин. В настоящей работе будет показано, как найти такое покрытие за линейное время.

Будем рассматривать простой неориентированный граф $G = (V, E)$, состоящий из n вершин и m ребер. Через $G - x$, где x – некоторая вершина, будем обозначать граф, полученный из G удалением вершины x и всех ребер, смежных с ней. Через $G - ke$ будем обозначать граф, полученный из G путем удаления любых (если не оговорено конкретно) k ребер. Через $G + ke$ будем обозначать граф, полученный из G путем добавления к нему любых (если не оговорено конкретно) k ребер. Так же будем придерживаться стандартных обозначений: K_n означает полный граф из n вершин, $K_{n,p}$ – полный двудольный граф из n вершин в первой доле и p вершин во второй, W_n – граф-колесо, состоящий из n вершин. Будем считать, что вместе с графом даны степени вершин, обозначаемые через $\deg[v]$, где v – некоторая вершина. Через $\delta(G)$ будем обозначать минимальную степень вершины, встречающуюся в G.

Алгоритм поиска трех вершин с квадратичной сложностью

В данном разделе рассмотрим простой алгоритм со сложностью $O(n^2)$, решающий поставленную задачу. Для этого понадобятся следующие теоремы:

Теорема 1 (Дирак) [4, 6]. Все 3-связанные графы в Ω_2 – это: $W_k (k \ge 3)$, $K_5, K_5 - e$, $K_{3,p}$, $K_{3,p} + e, K_{3,p} + 2e, K_{3,p} + 3e$. (В последних трех графах ребра добавляются в долю из трех вершин.)

Теорема 2 (Ловас) [4, 5]. Пусть G – мультиграф, не имеющий двух вершинно разобщенных циклов. Пусть $\delta(G) \ge 3$, и нет вершины, представляющей все циклы. Тогда верно одно из:

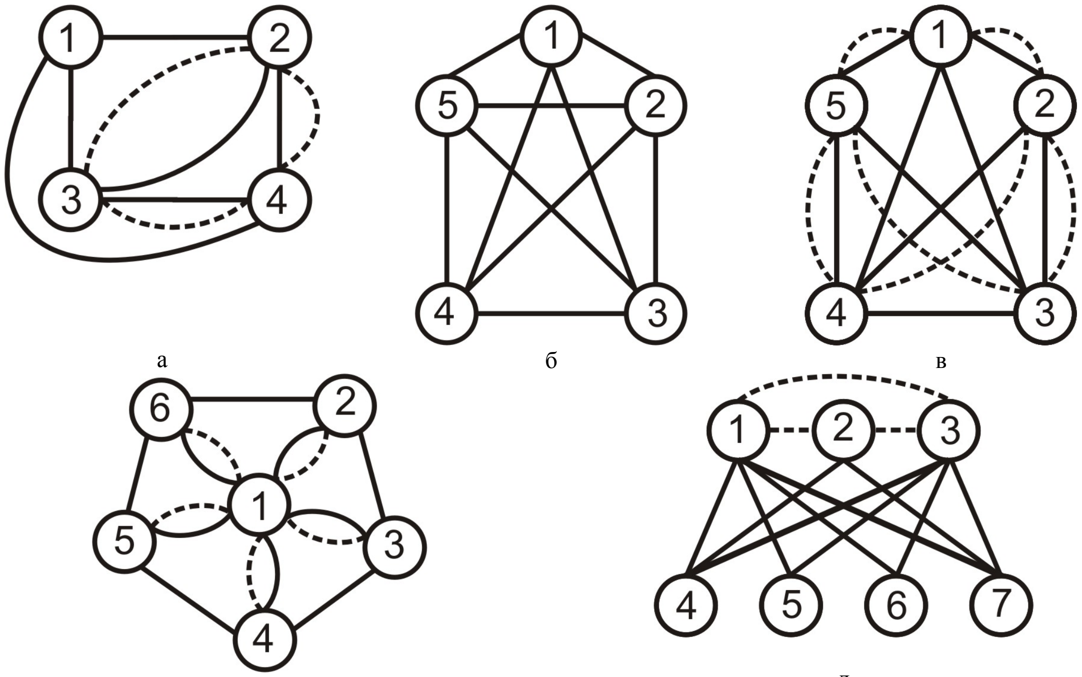

Рис. 1. Графы, не содержащие двух вершинно разобщенных циклов. Ребра, отмеченные штриховой линией, заменяются на несколько (в том числе и ноль) кратных ребер

1. G имеет 3 вершины;

2. $G = K_4$, один из треугольников может иметь кратные ребра (рис. 1а);

3. $G = K_5$ (рис. 1б);

4. $G = K_5 - e$, некоторые ребра, смежные с удаленным, могут быть кратными (рис. 1в);

5. G – колесо, спицы могут быть кратными (рис. 1г);

6. G получается из $K_{3,p}$ добавлением ребер (возможно, кратных) между вершинами первой доли (рис. 1д).

Теорема 3. Мультиграф G не содержит двух вершинно разобщенных циклов тогда и только тогда, когда $G - x$ – лес, для некоторой вершины $x \in V$ или G может быть получен из графа теоремы 2 подразбиением ребер или добавлением леса и не более одного ребра, соединяющего каждое дерево леса с этим графом.

Теперь на основе теоремы 3 построим алгоритм, проверяющий, имеет ли граф G два вершинно разобщенных цикла, и, в случае отрицательного ответа, выдающий не более трех вершин, представляющих все циклы.

Шаг 1. Посчитаем степень каждой вершины и будем удалять вершины степени 1 до тех пор, пока они есть, попутно корректируя степени смежных с ними вершин.

Шаг 2. Произведем операцию, обратную подразбиению ребер: будем стягивать ребра, инцидентные вершинам степени 2. При этом в графе могут образовываться кратные ребра или петли, запоминаем их. Получим граф G'.

Шаг 3. Проверим, есть ли у графа G' вершина, представляющая все циклы. Если такая вершина найдена, то стоп.

Шаг 4. Если ее нет, то граф G не содержит двух независимых циклов только тогда, когда G' – один из графов теоремы 2. Для каждого такого графа несложно выделить вершины, представляющие все циклы, например, следующие:

1. Если G состоит из трех вершин, то все три вершины.

2. Если $G = K_4$, то вершины с номерами 2 и 3.

3. Если $G = K_5$, то вершины с номерами 1, 2 и 3.

4. Если $G = K_5 - e$, то вершины с номерами 1, 2 и 5.

5. Если G – колесо, то вершины с номерами 1 и 2.

6. Если $G = K_{3,p}$, то вершины с номерами 1, 2 и 3.

Эти же вершины будут представлять все циклы графа G.

Дадим оценку сложности описанного алгоритма. По алгоритму одной из самых частых операций будет операция удаления вершины. Удалять вершину будем, удаляя все ребра, инцидентные ей, и помечая вершину как удаленную. Если хранить граф в виде обычных списков ребер, то операция удаления ребра будет выполняться за время $O(n)$. Модифицируем структуру хранения таким образом, чтобы удалять ребра за $O(1)$. Сделаем список двусвязным и добавим указатель на второе звено, содержащее то же самое ребро. Двусвязность списка позволит быстро (за $O(1)$) переопределять указатели на следующее и предыдущее звенья для звеньев, соседних с удаляемым, а третий указатель даст возможность за время $O(1)$ найти второе звено, соответствующее удаляемому ребру. Также алгоритму понадобятся степени вершин; будем считать, что они вместе с исходным графом являются входными данными.

Пусть исходный граф G содержит n вершин. Построим оценку сложности для каждого шага алгоритма:

Шаг 1. Удаление вершин степени 1. Каждая такая вершина содержит одно ребро, а значит, удаление вершины будет происходить за время $O(1)$. При удалении вершины будем проверять степень вершины v, смежной с ней, и, если она становится равной 1, также удалять v. Общая сложность данного шага будет равна $O(n)$.

Шаг 2. Стягивание ребер, инцидентных вершинам степени 2. Пусть есть вершина v степени 2 и смежные с ней вершины a и b. Стягивание ребра, инцидентного такой вершине, можно осуществить следующим образом: удалить вершину v и добавить ребро (возможно, кратное, или петлю при $a=b$) (a,b). Таким образом, операция стягивания ребра будет выполняться за время $O(1)$, и общая сложность всей процедуры будет $O(n)$.

Шаг 3. Поиск вершины, представляющей все циклы. Данную операцию можно выполнить за время $O(n^2)$ с помощью поиска в глубину.

Шаг 4. Сопоставление полученного графа с эталонными графами. Проверить получившийся граф на изоморфность с K_4, K_5, K_5-e можно за время $O(n)$. Для этого можно построить все возможные биекции между вершинами нашего графа и эталонного и проверить корректность расположения ребер. Степень каждой вершины в полученном графе не превосходит степени этой же вершины в исходном графе. Значит, если в полученном графе 4 или 5 вершин, то общее число ребер – $O(n)$, и каждая проверка будет выполняться за линейное время. Колесо можно охарактеризовать как граф, в котором вершина с самой большой степенью соединена со всеми остальными, и оставшиеся вершины образуют единственный цикл. Вершину v с максимальной степенью можно найти за время $O(n)$, перебрав все. Так как при выполнении шагов 1, 2 и 3 степени вершин не увеличивались, верно неравенство $\deg[v]<n$. Поэтому удалить v можно за $O(n)$, при этом проверяя, что v соединена со всеми остальными вершинами. Для того чтобы проконтролировать выполнение последнего условия, достаточно, по лемме о хороводах, показать, что все оставшиеся вершины имеют степень 2 и граф связен. Эти проверки можно выполнить за время $O(n)$. Последний вид графов имеет n' вершин, из которых n'–3 соединены только с оставшимися тремя. Вопрос принадлежности нашего графа к этому классу можно решить отдельно для $n'>6$ и для $n'\le 6$. Для $n'\le 6$ проверку выполняем так же, как для K_4, K_5, K_5-e. Для $n'>6$ выделяем все вершины степени 3 и проверяем, что останется ровно три вершины, с которыми соединены все выделенные. Такая проверка на принадлежность последнему классу графов также будет работать за время $O(n)$.

Таким образом, данный алгоритм работает за время $O(n^2)$.

Модификация алгоритма

Заметим, что из 4 шагов алгоритма только третий шаг имеет квадратичную сложность. Остальные выполняются за линейное от числа вершин время. Если бы удалось выполнить и третий шаг за $O(n)$, это бы привело к уменьшению общей сложности алгоритма. Покажем, как это можно сделать.

Пусть после работы первых двух шагов алгоритма получился мультиграф $G'=(V',E')$ с n' вершинами и m' ребрами, причем каждая вершина имеет степень хотя бы 3. Для простых графов есть условие существования цикла в графе: $m\ge n$. Если под циклом в мультиграфе понимать цикл, пару кратных ребер или петлю, то такая же теорема верна и для них. Назовем вершину v подозрительной, если $\deg[v]>m'–n'$. Таким образом, если вершина v не подозрительная, то при ее удалении в оставшемся графе всегда найдется цикл. Предположим теперь, что вершина v – подозрительная. Для того чтобы проверить, представляет ли она все циклы, удалим ее из графа и проверим оставшийся граф на ацикличность. Поскольку $\deg[v]<n$, удаление вершины v из графа можно выполнить за $O(n)$. Проверка на ацикличность выполняется с помощью поиска в глубину за время $O(n)$ [7]. Таким образом, проверить подозрительную вер-

шину на представление всех циклов можно за $O(n)$.

Теперь выделим два крайних случая, когда необходимо проверить не более двух вершин:

1. В графе есть вершина v с петлей. Поскольку петля есть цикл, проходящий только через v, то достаточно проверить только вершину v.

2. В графе есть вершина $v: \deg[v] \ge n'$. В этом случае у v есть кратные ребра, ведущие к вершине w. Значит, через одну из вершин v и w должны проходить все циклы.

Покажем, что проверить граф на выполнение этих случаев можно за время $O(n)$.

Рассмотрим первый случай. Исходный граф G был простым, а значит, петель там не было. Петли могли появиться только при выполнении шага 2 алгоритма в случае, когда $a = b$. Это легко отследить, причем добавление такой проверки не увеличит сложности второго шага. Кроме того, можно запомнить и вершину, на которой появилась петля. Таким образом, останется только проверить, будет ли найденная вершина представлять все циклы, что, как уже было показано, выполняется за $O(n)$.

Во втором случае несложно найти вершину v за время $O(n)$, перебрав все вершины. Поскольку для v выполняется неравенство $\deg[v] < n$, можно перебрать все инцидентные ей ребра и найти вершину w за время $O(n)$.

Если в результате обеих проверок не найдено кратных ребер и петель, то для любой вершины графа G' выполняется неравенство $3 \le \deg[v] < n'$. Для таких мультиграфов верна следующая теорема:

Теорема 4. Пусть дан мультиграф $G = (V, E)$, состоящий из n вершин и m ребер и не содержащий петель. Если для каждой вершины $v \in V$ выполняется неравенство $3 \le \deg[v] < n$, то в графе G не более пяти подозрительных вершин.

Доказательство. Предположим, граф G содержит $t > 2$ подозрительных вершин. Для каждой подозрительной вершины верно неравенство $\deg[v] > m - n$, просуммируем его по всем подозрительным вершинам:

$$\sum_{v:\deg[v]>m-n} \deg[v] > t(m-n).$$

Число ребер, инцидентных подозрительным вершинам, не превосходит m, следовательно,

$$2m > t(m-n),$$

или

$$m < n\frac{t}{t-2}.$$

Поскольку степень каждой вершины не меньше 3, верна оценка $m \ge \frac{3}{2}n$, откуда

$$\frac{3}{2} < \frac{t}{t-2}.$$

Из этого неравенства получаем оценку $t < 6$, что и требовалось доказать.

Подозрительные вершины имеют наибольшие степени, и проверку каждой вершины можно провести за время $O(n)$. Таким образом, *Шаг 3* алгоритма поменяется на *Шаг 3'*:

Шаг 3'. Выбрать вершину максимальной степени из еще не проверенных. Если она подозрительная, то проверить ее, иначе перейти на *Шаг 4*.

Список литературы

1. Bodlaender H. On disjoint cycles // International Journal of Foundation of Computer Science. 1994. V. 5. P. 230–238.

2. Erdos P., Posa L. On independent circuits contained in a graph // Canad. Journ. Math. 1965. V. 17. P. 347–352.

3. Diestel R. Graph Theory. Heidelberg, New York: Springer-Vergal, 2005. P. 44–46.

4. Bollobas B. Extremal graph theory. London: Academic Press, 1978. P. 110–119.

5. Lovasz L. On graphs not containing independent circuits (Hungarian) // MatLopak. 1965. V. 16. P. 289–299.

6. Dirak G. Some results concerning the structure of graphs // Canad. Math. Bull. 1965. V. 8. P. 459–463.

7. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ, 2-е издание: Пер. с англ. М.: Издательский дом «Вильямс», 2009.

A FEEDBACK VERTEX SET

M.V. Matrosov

It has been shown that the following problem can be solved in the time linear in the number of vertices: given a simple graph, find in it two vertex-disjoint cycles or build a feedback vertex set that contains no more than 3 vertices.

Keywords: graphs, disjoint cycles, cycle packing, cycle covering.