Студ. Б.З.ШАПИРО
(МИИТ)

АЛГОРИТМ ПОСТРОЕНИЯ ЦЕПНОГО ПОКРЫТИЯ НЕОРИЕНТИРОВАННОГО ГРАФА

Пусть $G=\{A,B\}$ – неориентированный граф ($card\,A=k$, $card\,B=n$). Известно, что лишь четное число его вершин может иметь нечетную степень. Обозначим его через 2ℓ. Известно также, что имеется ℓ ребернонепересекающихся цепей, содержащих в совокупности все ребра из B. Требуется указать алгоритм выбора этих цепей.

Эта задача может быть сведена к задаче построения эйлерова цикла в эйлеровом графе. Действительно, добавим ко множеству A еще одну вершину a_0, которую соединим со всеми вершинами, имеющими нечетную степень. В итоге получится граф $\tilde{G}$, степени вершин которого будут четными. Как известно, такой граф – эйлеров. Имея алгоритм построения в $\tilde{G}$ эйлерова цикла, начинающегося в вершине a_0, выпишем этот цикл. Отрезки цикла между местами, на которых стоит вершина a_0, дадут нам искомые ℓ цепей.

Ниже приводится один из алгоритмов построения эйлерова цикла в связном эйлеровом графе с оценкой числа операций

$$\vartheta \le k\left(k+\frac{n}{2}\right)+7{,}5n+2k,$$

где k – число вершин графа;
n – число ребер.

Алгоритм. Этап I. Шаг 0.1. Положим множество $A^0=\{a_0\}$, $A^1=\{\varnothing\}$, граф Γ^0 равным исходному эйлерову графу.

Шаг P.I. а) если $A^{p-1}=\{\varnothing\}$, то переход к шагу 0.П; б) если $A^{p-1}=\{a_1^{p-1},\dots,a_r^{p-1}\}$, то первой рассмотрим a_1^{p-1};

в) если рассматриваемая вершина a_i^{p-1} изолирована, то если $i<r$, рассмотрим вершину a_{i+1}^{p-1} и переход на начало в). В противном случае – переход к шагу P+1.1;

г) если a_i^{p-1} не изолирована, то осуществим следующую процедуру. Из вершины a_i^{p-1} ищется произвольная вершина, смежная с ней в имеющемся графе. Пусть это – вершина a_{j_1}. Ребро (a_i^{p-1},a_{j_1}) удаляется из графа. Из вершины a_{j_1} ищется произвольная вершина, смежная с a_{j_1} в имеющемся графе. Пусть это – вершина a_{j_2}. Ребро (a_{j_1},a_{j_2}) удаляется из графа. Номера вершин в порядке прохождения записываются. Процедура обрывается тогда, когда очередная найденная вершина после удаления ребра, инцидентного ей, окажется изолированной. Легко усмотреть, что по окончании процедуры выписанные вершины образуют цикл. Каждую из рассмотренных при построении цикла вершин в том случае, если она не принадлежит A^{p-1} и A^p включаем в A^p (один раз). Каждую из включенных в A^p вершин пометим (т.е. поставим им в соответствие вершину a_i^{p-1}). Назовем вершину a_i^{p-1} вершинной пометкой включенных вершин. Вершине a_i^{p-1} поставим в соответствие построенный цикл. Назовем этот цикл цикловой пометкой вершины a_i^{p-1}. Если $i<r$, рассмотрим вершину a_{i+1}^{p-1} и переход на в). Если $i=r$, то осуществим переход к шагу P+1.1.

Этап П. Шаг 0.П. Возьмем для рассмотрения вершину a_0.

Шаг 1.П. Очередным элементом цикла запишем рассматриваемую вершину. Если вершина впервые появилась на этапе П, то перейдем к шагу 4.П. Если она уже рассматривалась, то перейдем к шагу 2.П.

Шаг 2.П. Выясним, является ли рассматриваемая вершина последней в цикле, построенном на этапе 1, из которого она была выписана: если является, то перейдем к шагу 3.П; если не является, то для рассмотрения выбираем следующую за ней вершину того же цикла этапа 1. Переход на шаг 1.П.

Шаг 3.П. Проверим, является ли данная вершина вершиной a_0: если является, то переходим на конец работы алгоритма; если не является, то ищем вершинную пометку рассматриваемой вершины. У вершины, являющейся вершинной пометкой, ищем цикловую пометку. Найдя этот цикл, осуществим перебор его вершин от начала к концу, пока не обнаружим рассматриваемую вершину. Для рассмотрения возьмем следующую вершину этого цикла. Переход на шаг 1.П.

Шаг 4.П. Пометим, что вершина появилась на этапе П. Выясним, есть ли у вершины цикловая пометка: если есть, то переход на шаг 5.П; если отсутствует, то следующей рассмотрим идущую за ней вершину того же цикла этапа 1. Переход на шаг 1.П.

Шаг 5.П. Найдя соответствующий цикл, рассмотрим в нем вторую от начала вершину. Переход на шаг 1.П.

Докажем, что последовательность вершин, выписанных на этапе П, есть искомый эйлеров цикл. Действительно, на этапе 1 строятся некоторые циклы, причем все они являются ребернонепересекающимися и содержат в совокупности все ребра исходного графа.

Это следует из того, что, во-первых, если вершина попала в множество A^p, то на шаге P+1.1 она станет изолированной и, во-вторых, в множество A^p на шаге P.1 попадут, по крайней мере, все еще не рассмотренные на этапе 1 вершины, смежные с A^{p-1}. Изолируя вершины из A^{p-1}, мы все их рассмотрим. Отсюда и из связности графа получается, что число шагов этапа 1 не превосходит $d+1$, где d – максимум из расстояний от вершины a_0 до всех остальных вершин исходного графа.

Далее, начав выписывать последовательность вершин некоторого цикла, построенного на этапе 1, мы осуществим этот процесс до конца на этапе П, прерываясь на выписывание вставок, которые являются циклами, начинающимися в вершинах выписываемого цикла.

В самом деле, для циклов, построенных на этапе 1 и последнем его шаге, это очевидно. На последнем шаге не появляется новых вершин, следовательно, согласно алгоритму, циклы этапа 1 выписываются на этапе П без вставок от начала до конца. Предположим, что обсуждаемый факт верен для циклов, построенных на шагах от m +1.1 до последнего шага этапа 1 включительно. Докажем его для циклов, построенных на шаге m .1. При выписывании цикла шага m .1 на этапе П мы можем натолкнуться на новую вершину. Если это произойдет, то, согласно индуктивному предположению, мы выпишем вставку и продолжим выписывание цикла шага m .1, вернувшись в ту же самую вершину, что и приведет к успеху в силу конечности числа вставок.

Наконец, на этапе П мы начнем выписывать все циклы, построенные на этапе 1.

Действительно, этап П начнется выписыванием цикла, построенного на шаге 1.1. Тем самым мы натолкнемся на все вершины из A^1, так как цикл будет пройден от начала до конца. Следовательно, мы начнем выписывать все циклы шага 2.1. Так как мы выпишем их до конца, то натолкнемся на все вершины из A^2 и т.д.

Оценка числа операций алгоритма. Пусть K – число вершин связного эйлерового графа, n – число его ребер.

Этап 1. На поиск нового ребра уходит $z/2+1$ операций, где z – степень вершины. Если считать, что на поиск ребра в ЭВМ уходит K операций, то всего затрачивается $(\frac{n}{2}+K)K$ операций.

На выписывание циклов, вычеркивание ребер, установление пометок затрачивается всего $\frac{n}{2}+\frac{2n}{3}+n+\frac{n}{2}+K+\frac{n}{2}$ операций. Вершина входит в циклы этапа 1 $z/2$ раз, если она не находится в начале цикла. Если она находится в начале, то она записывается еще один раз. При этом нужно учесть, что число циклов этапа 1 меньше или равно $n/3$ (в одном цикле больше либо равно 3 ребрам). Вычеркиваются ребра в графе n раз. Каждая вершина проверяется на наличие пометки $^2/_2$ раз и один раз помечается.

Выбор множества A^p осуществляется за $n/2 + K$ операций, так как $^2/_2$ раз проверяется, использовалась ли вершина раньше; один раз она выписывается в A^p.

Этап П. На запись и проверку, появляется ли вершина в первый раз на этапе П и является ли она вершиной a_0 уходит $2n$ операций.

На поиск цикловых пометок и установление номера следующей вершины уходит не более n операций.

Итак, всего не более $K(K+n/2)+7{,}5n+2K$ операций. Зависимость линейна по n и квадратична по K.

Ст.научн.сотр. М.М.ГЕЛЛЕР
(МИИТ)

ОБ ОДНОМ МЕТОДЕ ВЫБОРА СТРАТЕГИИ ЭВРИСТИЧЕСКОГО ПОИСКА

При решении значительного числа комбинаторных задач большой размерности используются алгоритмы эвристического перебора по дереву состояний в поисках оптимального решения. Но во многих случаях общее количество путей по дереву очень велико и нет никакой реальной возможности просмотреть все пути. Следует каким-то образом ограничить количество просматриваемых вершин дерева. С другой стороны, очень сильно уменьшить их количество не целесообразно, так как тогда будет мала вероятность найти не только наилучшее решение, но и те решения, которые не очень значительно отличаются от него в смысле выбранного критерия качества. Напомним, что качество решений определяется оценочной функцией (критерием), т.е. каждой вершине дерева ставится в соответствие некоторое число. Это число, как правило, определяется большим количеством факторов. Так как заранее учесть влияние всех этих факторов не представляется возможным, имеет смысл рассматривать оценки вершин (значения оценочной функции) как случайные величины. Проведя статический анализ по поддеревьям данного дерева, можно найти оценку функции распределения критерия как случайной величины.

Количество рассматриваемых вершин зависит от стратегии перебора по дереву. Понятно, что нужно сравнить между собой именно стратегии.