УДК 519.1

О задачах обхода графа

Т. Е. Бояринцева, А.А. Мастихина

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

В статье рассматривается тема соотношения «наглядного» способа изложения действий на графах (с использованием рисунка) и «абстрактного» (опирающегося на представление графа посредством матрицы). Такого рода проблема (изложение наглядных действий при помощи инструмента дискретной математики) нередко возникает в преподавании предмета. Для задачи построения матрицы достижимости и определения количества и состава компонент связности даются два алгоритма решения. В качестве примера описания графом системы с различными возможными состояниями приводится задача о переливании. Для другого примера графической задачи дается решение, которое обосновывается уже с применением булевых функций. Также рассматривается задача о построении гамильтонова цикла, связанного с обходом полей шахматной доски фигурой коня.

Ключевые слова: *графы, обходы, матрица смежности, достижимость, эйлеров цикл, булевы функции.*

Введение. Преподавание дискретной математики студентам технических специальностей зачастую связано с определенными трудностями.

Многие специальности обходятся без углубленного изучения алгебраических структур, необходимых для основательного подхода к предмету дискретной математики. А все сужающиеся временные границы не позволяют как следует освоить эту область уже в рамках дискретной математики.

В то же время большинство задач имеет практический смысл, что, с одной стороны, упрощает, но, с другой, — усложняет преподавание. Ведь многие частные случаи распространенных задач могут решаться и без применения теоретических знаний.

Особенно это относится к теории графов. Само понятие графа двояко. Граф интуитивно понимается как рисунок (геометрический граф). В то же время математическое его определение — как пары множеств — уясняется с трудом. Однако для комплексного решения классов задач используются алгоритмы, реализация которых требует абстрактного представления графов.

В данной работе рассматривается соотношение наглядного и абстрактного способов представления действий с графами в области задач, связанных с обходами.

Будет подробно разобрана задача определения достижимости вершин. Одно из ее приложений — нахождение эйлерова цикла в графе — будет рассмотрено в первую очередь. Если графом задается некоторая система, которая может находиться в разных состояниях (вершины — состояния, ребра — возможные переходы), то достижимость означает, что можно попасть из одного фиксированного состояния в другое. Рассматривается сведение подобных задач к операциям с матрицами.

Второй раздел статьи имеет цель привнести элемент занимательности в преподавание предмета. Все необходимые сведения из теории графов и теории булевых функций содержатся в [1–4].

1. Компоненты связности графа; эйлеровы циклы. Напомним, что цикл в графе называется эйлеровым, если он проходит через каждое ребро ровно один раз; если такой цикл существует, то граф называется эйлеровым. Он характеризуется тем, что все его вершины имеют четную степень. Исторический обзор задачи поиска эйлерова цикла приведен в [5].

Следующий алгоритм Флери дает эйлеров цикл.

Исходя из произвольно выбранной вершины, идем по ребрам, причем по мосту — только тогда, когда это неизбежно. Напомним, что мост — это ребро, удаление которого увеличивает число связных компонент.

Пройденные ребра за собой стираем.

Алгоритм заканчивает работу, когда ребер больше нет.

В таком виде алгоритм Флери не годится для реализации на ЭВМ. Опишем матричную реализацию алгоритма Флери; она может быть выполнена на ЭВМ. Для этого построим матрицу достижимости графа, т. е. матрицу, в которой единицы на пересечении i-й строки и j-го столбца означают достижимость i-й вершины из j-й.

Пусть A — матрица смежности графа G, в которой на главной диагонали стоят единицы. Такую матрицу смежности можно считать матрицей достижимости не более чем за один шаг — на пересечении i-й строки и j-го столбца стоит единица, если i-я и j-я вершины соединены ребром, и каждая вершина достижима из себя же. Используем логическое умножение матриц (матрицы перемножаются как обычно, но в ходе вычислений все положительные числа заменяются на единицу). То есть при таком перемножении матриц $n \times n$ результирующая матрица $C = A \cdot B$ получается по правилу $c_{ij} = \max_{k=1,\dots,n} \{a_{ik} b_{kj}\}$. Последовательно находим степени A, A^2, …, применяя описанное умножение. Единицы в k-й степени матрицы будут означать существование пути длины k. Например, матрица A^2 — матрица достижимости не более чем за два шага (причем отличные от нуля слагаемые при перемножении i-й строки и j-го столбца соответствуют существующим вариантам путей не более чем из двух ребер, соединяющих вершины i и j). Работа заканчивается на матрице A^k (с наименьшим k, для которого $A^k = A^{k+1}$). Эта матрица есть матрица достижимости графа G. (Отношение достижимости является рефлексивно-транзитивным замыканием отношения смежности.)

Строки этой матрицы разбиваются на массивы, каждый из которых состоит из одинаковых строк. Соответствующие вершины графа дают разбиение на компоненты связности.

Пример 1. Пусть граф задан матрицей смежности

$$A = \begin{Vmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{Vmatrix}.$$

Тогда

$$A^2 = \begin{Vmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{Vmatrix}, \quad A^3 = \begin{Vmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{Vmatrix}.$$

Выясняется, что матрица $A^3 = A^2$, и, следовательно, равна матрице достижимости. Матрица достижимости содержит три одинаковые строки — первую, третью и четвертую, соответствующие вершины образуют одну компоненту связности, а другую образует вторая вершина. Геометрический граф для этой матрицы изображен на рис. 1.

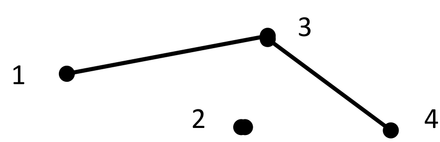

Рис. 1. Граф с матрицей смежности A

Также для нахождения матрицы достижимости можно использовать другую, более быструю, процедуру.

Матрица достижимости будет строиться из матрицы смежности с единицами на главной диагонали.

Последовательно просматривается первая строка матрицы A. Если обнаруживается единица в i-м столбце, то все единицы i-й строки переносятся в первую строку той же матрицы на соответствующие места (если там уже стоит единица, то она остается), а номер i заносится в специальный массив достижимых вершин D_1 (будущий список вершин из одной компоненты связности). Далее ищем в первой строке единицы после i-го столбца. Дойдя до конца первой строки, возвращаемся в начало и повторяем процедуру (не обращая внимания на столбцы с номерами из D_1) до тех пор, пока в этой первой строке не перестанут появляться новые единицы. В итоге мы получим список D_1 вершин, достижимых из первой вершины, т. е. мы получим состав компоненты связности, содержащей первую строку. Строки с номерами, попавшими в список достижимых вершин, можно заменить на получившуюся первую строку. Далее ищем строку, номер которой не попал в этот список, и для нее проделываем ту же процедуру, которая уже описана для первой строки. Снова получаем список компоненты связности. Так поступаем до тех пор, пока все вершины не окажутся в каком-либо из найденных списков. Количество списков есть количество компонент связности.

Пример 2. Пусть граф задан матрицей смежности

$$A = \begin{Vmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{Vmatrix}.$$

Первая единица первой строки не меняет строку, следующая единица встречается в пятом столбце. Первая строка становится (1 1 0 0 1), а $D_1 = \{1, 5\}$. Просматривая первую строку сначала, доходим до единицы во втором столбце. Теперь $D_1 = \{1, 5, 2\}$, а первая строка не изменилась от добавления единиц второй строки. Новых единиц нет, компонента связности найдена. Повторяем процедуру для третьей строки, $D_3 = \{3, 4\}$. Компоненты связности получены.

Эта же процедура нахождения матрицы достижимости позволяет выяснить, является ли данное ребро мостом. Достаточно удалить это ребро и для полученного нового графа найти (тем же способом) компоненты связности. Ребро будет мостом лишь в том случае, если при переходе к новому графу число компонент связности увеличилось.

В алгоритме Флери приходится многократно выяснять, является ли очередное «испытуемое» ребро мостом. Выбираем ребро и пробуем его удалить, стабилизируем матрицу смежности, анализируем полученное число компонент связности. Если оно возросло, то восстанавливаем удаленное ребро и ищем другое из той же вершины. Мы видим, что алгоритм Флери может сводиться к матричным вычислениям и в таком модифицированном виде годится для реализации на ЭВМ.

При аудиторном рассмотрении данной задачи можно напомнить студентам о популярной задаче «нарисуй, не отрывая карандаша». Эта задача может быть решена с помощью теоремы Эйлера и алгоритма Флери. Собственно, большинство интуитивно решает эту задачу правильно, начиная рисовать из точки, где сходится нечетное количество линий, и при движении стараются не оставлять изолированными фрагменты рисунка.

Большое количество задач поиска эйлерова цикла содержится в [6].

2. В данном разделе будут рассмотрены две задачи о достижении системой определенного состояния. Для задачи 2.1 [7] будет разобрано решение, использующее графы [7], для 2.2 — булевы функции. Также приведена задача об обходе шахматной доски.

2.1. Задача о переливании. Имеется три кувшина с водой: один объемом 8 л, заполненный водой, и два пустых, объемом 5 и 3 л. Требуется разделить 8 л пополам, т. е. получить две емкости по четыре литра в каждой.

Решение задачи с помощью графов.

Будем обозначать каждое состояние системы из трех кувшинов парой чисел (a, b), обозначающих количество литров в кувшинах емкостью соответственно 5 и 3 л. На самом деле, полное описание — это тройка (a, b, c), но содержимое третьего кувшина c (объемом 8 л) определяется условием $a + b + c = 8$. Числа a, b, c — натуральные.

Начальное состояние системы — (0, 0). Желаемое состояние — (4,0). Другие варианты не подходят, так как $b \leq 3$.

Пары (a, b) будут вершинами графа. Ориентированными ребрами будем обозначать возможный переход из одного состояния в другое с помощью однократного переливания.

Вопрос сводится к тому, достижима ли вершина (4,0) из вершины (0,0) в таком графе и каким образом. Рассмотрим подробнее, как устроен граф.

Из вершины не может выходить более шести ребер. Действительно, состояние (a, b, c) может измениться, если перелить воду, например, из первого кувшина во второй или третий. Причем в зависимости от количества воды в другом кувшине можно либо вылить в него всю воду, либо долить доверху, и тогда в первом кувшине еще останется вода.

Если перелить воду из первого кувшина во второй, то состояние будет $(0, a + b, c)$ при $a + b \leq 3$ или $(a - 3 + b, 3, c)$ при $a + b > 3$, а если в третий кувшин, то $(0, b, c + a)$ или $(a - 8 + c, b, 8)$. Аналогично возможны, самое большее, по два изменения при переливе воды из второго и третьего кувшинов. Но не из каждой вершины выходит ровно шесть ребер, возможны следующие случаи:

- 4 ребра, если один из меньших кувшинов полон ($a = 5$ или $b = 3$) или пуст ($a = 0$ или $b = 0$);
- 3 ребра, если один кувшин пуст, а другой полон (это случаи (5, 0) или (0, 3));
- 2 ребра, если меньшие кувшины оба полны или оба пусты (это (0, 0) и (5, 3));

Вершины можно изображать как точки на плоскости с координатами (a, b).

На рис. 2 изображен граф с вершинами, достижимыми из (0,0). Ребро со стрелками на обоих концах означает пару ребер с проти-

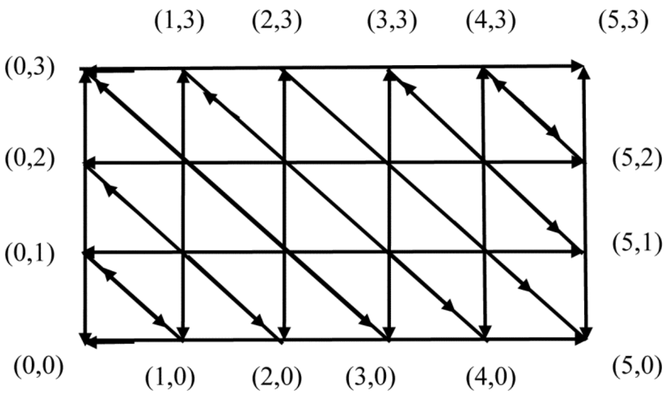

Рис. 2. Граф возможных изменений состояний системы

воположными направлениями, т. е. одним переливанием можно перейти из одного состояния в другое и обратно.

Поиском достижимых вершин можно найти несколько путей, ведущих в (4,0), а также установить, что кратчайшим из них является путь

$$(0,0)\to(5,0)\to(2,3)\to(2,0)\to(0,2)\to(5,2)\to(4,3)\to(4,0).$$

Также видно, что далеко не все вершины с целыми точками достижимы из (0,0), и ситуация изменится, если начальное состояние будет какое-либо другое.

2.2. Игра «Открой окна». На экране вашего компьютера — прямоугольник размером $m \times n$, разбитый (отрезками прямых) на квадратики размером 1×1; квадратики назовем окнами, а множество всех окон обозначим через A. Таким образом, A состоит из mn окон. Для любого окна $a \in A$ обозначим через T_a множество окон, находящихся на той же горизонтали либо на той же вертикали. (Это множество есть либо крест, либо буква *Т* (возможно, урезанная), либо буква *Г* (она также может оказаться урезанной). Некоторые окна закрыты (они помечаются крестиком), остальные открыты.

Начинается игра. Подводим курсор к какому-либо окну $a \in A$ и нажимаем на мышь. При этом состояние всех окон, входящих в множество T_a, меняется на противоположное (открытые закрываются, закрытые открываются). Задача состоит в том, чтобы несколькими нажатиями открыть все окна. Вряд ли вы добьетесь успеха, действуя наугад. Правильная стратегия состоит в следующем.

Предварительно введем одно определение. Пусть на некотором этапе игры на экране появляется конфигурация, состоящая из закрытых окон. Возьмем произвольное окно $a \in A$. Назовем его нечетным, если число закрытых окон в множестве T_a нечетно. Теперь стратегия описывается крайне просто: **на каждом этапе ваших действий нажимайте на нечетное окно; если таких окон несколько, нажимайте на любое.**

Приведем обоснование указанного алгоритма; оно опирается на булевы функции. Рассмотрим множество Z_2, состоящее из двух булевых величин 0, 1 с булевыми операциями: обычное умножение и сложение по модулю 2. Напомним, что на языке алгебры множество Z_2 есть поле. Возьмем далее множество всех функций $a \to f(a)$, где a пробегает множество окон, а значения функции принадлежат множеству Z_2. Для таких функций определены операция сложения по модулю 2 и операция умножения на элементы множества Z_2. Другими словами, получили линейное пространство L над полем Z_2. Его размерность равна mn (в качестве базиса можно взять набор функций, каждая из которых принимает значение 1 только на одном окне, а на остальных окнах принимает значение 0). Понятия линейной зависимости, базиса и другие вводятся так же, как в обычной линейной алгебре (над полем вещественных либо комплексных чисел).

Определена и операция поточечного перемножения двух функций.

Введем булево скалярное произведение двух функций по формуле $\left(f_1 f_2\right) = \sum_a f_1(a) f_2(a)$. Итак, перемножаем функции f_1, f_2 и берем сумму (по модулю 2) значений полученной функции для всех $a \in A$.

Повторяя общеизвестные понятия линейной алгебры, назовем функции ортогональными, если $\left(f_1, f_2\right) = 0$. Набор функций f_k назовем ортонормированным, если $\left(f_i, f_k\right) = 0$ при $i \neq k$ и $\left(f_k, f_k\right) = 1$ для всех k.

Для каждого окна a обозначим через f_a функцию, которая принимает значение 1 на окнах, входящих в множество T_a, и значение 0 — на всех других окнах (другими словами, f_a есть характеристическая функция множества T_a).

Лемма. *Семейство функций f_a есть ортонормированный базис пространства L.*

Читатель без труда докажет эту лемму (следует проследить, в каком месте доказательства используется то, что m, n — четные числа).

Вернемся к задаче 2.2 и обозначим через E множество закрытых окон и через g_E характеристическую функцию этого множества. Таким образом, действия над множествами превращаются в действия над характеристическими функциями. Важнейшее обстоятельство состоит в следующем: изменить состояние всех окон, входящих в множество T_a, равносильно замене функции g_E функцией $g_E \oplus f_a$.

Разложим функцию g_E в линейную комбинацию функций f_a, обозначив коэффициенты разложения через $r(a)$. Коэффициенты $r(a)$ находим по очевидным правилам вычисления коэффициентов ряда Фурье по ортонормированному базису. Таким образом, $r(a) =$

$=\left(g_E, f_a\right)$. Отсюда видно, что $r(a) = 1$ лишь тогда, когда пересечение множеств E и T_a состоит из нечетного множества квадратиков.

Имеем равенство $g_E \oplus \sum_a r(a) f_a = 0,$ из которого следует обещанное обоснование алгоритма решения задачи 2.2.

2.3. Конь на шахматной доске. Приведем еще один пример занимательной задачи. Это один из красивых примеров задачи построения гамильтонова цикла в графе. Гамильтоновым циклом называется цикл, проходящий через все вершины графа.

В общем случае построение гамильтонова цикла решается, например, с помощью алгоритма «поиска в глубину» [1], но в частном случае бывают необычные решения. Пусть имеем граф на 64 вершинах, которые сопоставим клеткам шахматной доски, и две вершины считаются смежными, если из одной в другую можно попасть одним ходом шахматного коня. Задача состоит в том, чтобы конем обойти все клетки шахматной доски, побывав на каждой клетке только по одному разу, и вернуться в исходное положение.

Маршрут Яниша. Начинаем с середины доски, позиция 1, далее посещаем клетки с числами 2, 3, ..., 64 (по ходу коня).

50	11	24	63	14	37	26	35
23	62	51	12	25	34	15	38
10	49	64	21	40	13	36	27
61	22	9	52	33	28	39	16
48	7	60	1	20	41	54	29
59	4	45	8	53	32	17	42
6	47	2	57	44	19	30	55
3	58	5	46	31	56	43	18

Отметим интересное свойство этого маршрута: при повороте доски на 180° первая половина маршрута (номера с 1 до 32) переходит во вторую (номера с 33 по 64).

Также приведем другой пример маршрута, примечательный тем, что для него существует мнемоническое стихотворение, сочиненное гроссмейстером Василием Пановым (впервые опубликовано в журнале «Наука и жизнь», [8]). Каждая строчка этого стихотворения содержит две пары слов. Каждой паре соответствует поле шахматной доски: первая буква первого слова — столбец (буква в русском произношении), первая буква второго — строка (цифра). Например: Алеет Осень — A1, Ценными Дарами — C2, Еще Один — E1, Животворящий День — G2 и т. д.

	A	B	C	D	E	F	G	H
1	1	50	63	22	3	36	61	24
2	52	21	2	35	62	23	4	37
3	49	64	51	20	43	16	25	60
4	32	53	14	17	34	19	38	5
5	11	48	33	42	15	44	59	26
6	54	31	10	13	18	41	6	39
7	47	12	29	56	45	8	27	58
8	30	55	46	9	28	57	40	7

Заключение. Рассмотрены алгоритмы нахождения матрицы достижимости графа, не требующие глубоких теоретических знаний, но подводящие к математически строгому пониманию задачи. Приведено несколько примеров, которые можно разбирать на практических занятиях. Продемонстрирован плавный переход от решения наглядных задач угадыванием к разумному перебору и, далее, к нахождению и обоснованию аналитического алгоритма решения.

ЛИТЕРАТУРА

[1] Белоусов А.И., Ткачев С.Б. *Дискретная математика*. Москва, Изд-во МГТУ им. Н.Э. Баумана (серия математика в техническом университете, 2001, вып. XIX).

[2] Уилсон Р. *Введение в теорию графов*. Москва, Мир, 1977.

[3] Яблонский С.В. *Введение в дискретную математику*. Москва, Наука, 1979.

[4] Нефедов В.Н., Осипова В.А. *Курс дискретной математики*. Москва, Изд-во МАИ, 1992.

[5] Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. *Теория графов*. Москва, Наука, 1990.

[6] Гаврилов Г.П., Сапоженко А.А. *Задачи и упражнения по курсу дискретной математики*. Москва, Наука, 1992.

[7] Оре О. *Теория графов*. Москва, Наука, 1968.

[8] Панов В.Н. Шахматы и мнемотехника. *Наука и жизнь*, 1969, № 5.

Статья поступила в редакцию 28.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Бояринцева Т.Е., Мастихина А.А. О задачах обхода графа. *Инженерный журнал: наука и инновации*, 2013, вып. 5. URL: http://engjournal.ru/catalog/pedagogika/ hidden/735.html

Мастихина Анна Антоновна родилась в 1986 г., окончила механико-математический факультет МГУ им. М.В. Ломоносова в 2008 г., канд. физ.-мат. наук, ассистент кафедры «Высшая математика» МГТУ им. Н.Э. Баумана. Автор научных статей по теории автоматов, а также методических работ.
e-mail: anmast@yandex.ru

Бояринцева Татьяна Евгеньевна родилась в 1962 г., окончила факультет прикладной математики МИЭМа в 1986 г., канд. физ.-мат. наук, доц. МГТУ им. Н.Э. Баумана. Автор ряда статей по гидродинамике и учебных пособий по математике для студентов. e-mail: t.bojare@mail.ru