УДК 519.172.3

© 2017 г. **Я.Н. Гусеница**, канд. техн. наук
(Военно-космическая академия имени А.Ф. Можайского, Санкт-Петербург)

АЛГОРИТМ ПОИСКА ЗОН В УПРАВЛЯЮЩИХ ГРАФАХ

В работе затрагивается задача поиска зон в управляющих графах. Проанализированы известные алгоритмы, разработанные для решения данной задачи. Предложен новый алгоритм, который в отличие от известных решений является более простым в программной реализации и имеет меньшую трудоемкость.
Ключевые слова: управляющий граф, зоны, алгоритм.

DOI: 10.22250/isu.2017.53.119-124

Введение

В настоящее время для моделирования систем различного назначения многие отечественные и зарубежные исследователи очень часто прибегают к применению теории графов. Этим объясняется значительное появление все больших разновидностей графов, обладающих уникальными свойствами и требующих оригинальных алгоритмов для их анализа.

Одной из разновидностей графов является управляющий граф, с помощью которого зачастую представляют алгоритм работы как программных, так и аппаратных систем.

Под управляющим графом (графом переходов, р-графом) понимают ориентированный граф $G(V,E)$, обладающий следующими свойствами [1]:

1) отсутствие параллельных дуг;

2) в множестве V вершин графа выделена вершина s – вход графа;

3) в множестве V вершин графа выделена хотя бы одна вершина f – выход графа;

4) каждая вершина $v \in V$ достижима из входа s;

5) каждая вершина $v \in V$ достигает выход f.

Важной задачей исследования управляющих графов является выявление различных фрагментов некоторого вида: зон, интервалов, гамаков и т.д. [2].

Зоной называется подграф $C_j = (X_j, U_j)$ управляющего графа G, в котором для любой пары вершин найдется путь из одной в другую.

Для отыскания зон необходимо реализовать два этапа. На первом этапе находят сильно связанную компоненту (бикомпоненту), т.е. зону, максимальную относительно включения вершин. Обычно для этого используют различные алгоритмы, – например, Евстигнеева, Лейфмана, Карзанова, Касьянова, Тарьяна, Фараджева или матричный алгоритм, наименьшая трудоемкость которых составляет $O(\max(n,m))$, где n – количество вершин, а m – количество дуг управляющего графа [3 – 8]. При этом перечисленные алгоритмы основаны на обходе управляющего графа, представленного списками смежности, с использованием стратегии поиска в глубину. На втором этапе отыскивают последовательность вложенных зон на основе матричного алгоритма, описанного в [1] и имеющего трудоемкость $O(n^3)$. Таким образом, решение задачи поиска зон занимает трудоемкость $O(\max(n,m)) + O(n^3)$.

В настоящей работе предлагается новый алгоритм, который основан на обходе управляющего графа, представленного матрицей смежности, с использованием стратегии поиска как в глубину, так и в ширину. Разработанный алгоритм является более простым в программной реализации и имеет трудоемкость не более $O(n^3)$.

Содержание алгоритма

Для реализации стратегии поиска в глубину в предлагаемом алгоритме в качестве исходных данных необходима B-нумерация вершин управляющего графа. B-нумерацией вершин управляющего графа G называется инъективное отображение множества V вершин управляющего графа G в множество натуральных чисел при поиске в ширину. При этом поиском в ширину называется регулярный обход управляющего графа, который осуществляется по следующему правилу:

находясь в вершине v_i, необходимо двигаться в смежную ей и ранее не пройденную вершину (если таковая найдется), запоминая дугу, которая впервые к ней привела;

если из вершины v_i невозможно попасть в смежную ей и ранее не пройденную вершину или таковой вообще не существует, то необходимо вернуться в вершину v_{i-1} и продолжить поиск в ширину.

В свою очередь, при реализации стратегии поиска в ширину в предлагаемом алгоритме необходимо применение M-нумерации вершин управляющего графа. M-нумерацией вершин управляющего графа G называется инъективное отображение множества V вершин управляющего графа G в множество натуральных чисел при поиске в глубину. При этом поиском в глубину называется регулярный обход управляющего графа, который осуществляется по следующему правилу:

находясь в вершине v_i, необходимо двигаться в любую другую, ранее не пройденную вершину (если таковая найдется), запоминая дугу, которая впервые к ней привела;

если из вершины v_i невозможно попасть в ранее не пройденную вершину или таковой вообще не существует, то необходимо вернуться в вершину v_{i-1} и продолжить поиск в глубину.

Кроме того, для выполнения алгоритма необходимы матрица смежности вершин $A(G)$ с M-номерами и матрица достижимости вершин $R(G)$ с M-номерами. Под матрицей смежности $A(G)$ управляющего графа G с n вершинами понимается квадратная матрица порядка n с элементами a_{ij}, которые принимают значение 1, если вершина v_i и v_j смежные, и 0 – в противном случае. Соответственно матрицей достижимости $R(G)$ управляющего графа G с n вершинами называется квадратная матрица порядка n с элементами r_{ij}, которые принимают значение 1, если вершина v_j достижима из v_i, и 0 – в противном случае.

Алгоритм, показанный на рис. 1, основан на выполнении следующих шагов.

Шаг 1. Ввод исходных данных.

Шаг 2. Установка значения счетчика $i = 0$.

Шаг 3. Проверка значения счетчика i. Если $i = n$, то необходимо перейти на шаг 12.

Шаг 4. Увеличение значения счетчика $i = i + 1$.

Шаг 5. Установка значения счетчика $j = 0$.

Шаг 6. Проверка значения счетчика j. Если $j = n$, то следует перейти на шаг 3.

Шаг 7. Увеличение значения счетчика $j = j + 1$.

Шаг 8. Проверка значения элемента a_{ij} матрицы смежности $A(G)$ и элемента r_{ij} матрицы достижимости $R(G)$. Если не выполняется условие, что $a_{ji} = 1$ и $r_{ij} = 1$, то необходимо перейти на шаг 6.

Шаг 9. Если не выполняется условие, что M-номер j-й вершины больше M-номера i-й вершины и B-номер j-й вершины больше либо равен M-номеру j-й вершины, то следует перейти на шаг 6.

Шаг 10. Установка очередного номера зоны $z = z + 1$.

Шаг 11. Сначала помещение в зону таких вершин, что $\forall l : l \in \left[B_i, B_j\right]$. Затем переход на шаг 6.

Шаг 12. Вывод результатов.

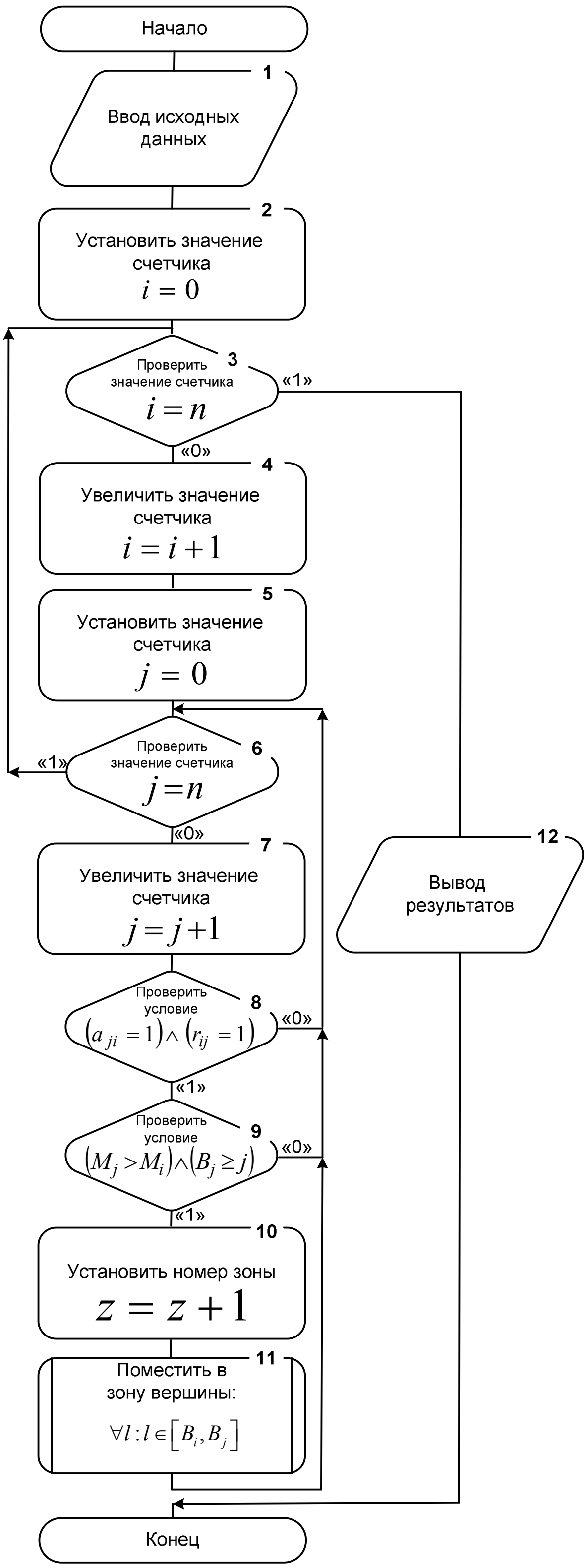

Рис. 1. Схема алгоритма поиска зон в управляющих графах.

Результаты работы алгоритма

Для наглядности рассмотрим результат работы алгоритма на примере. В качестве анализируемого управляющего графа G выбран алгоритм обнаружения пачек двоично-квантованных сигналов, представленный в [9]. Данный управляющий граф состоит из 64 вершин и 81 дуги. Его графическое изображение представлено на рис. 2, а матрица достижимости с произвольной M- и B-нумерациями вершин указана в таблице через наклонную черту.

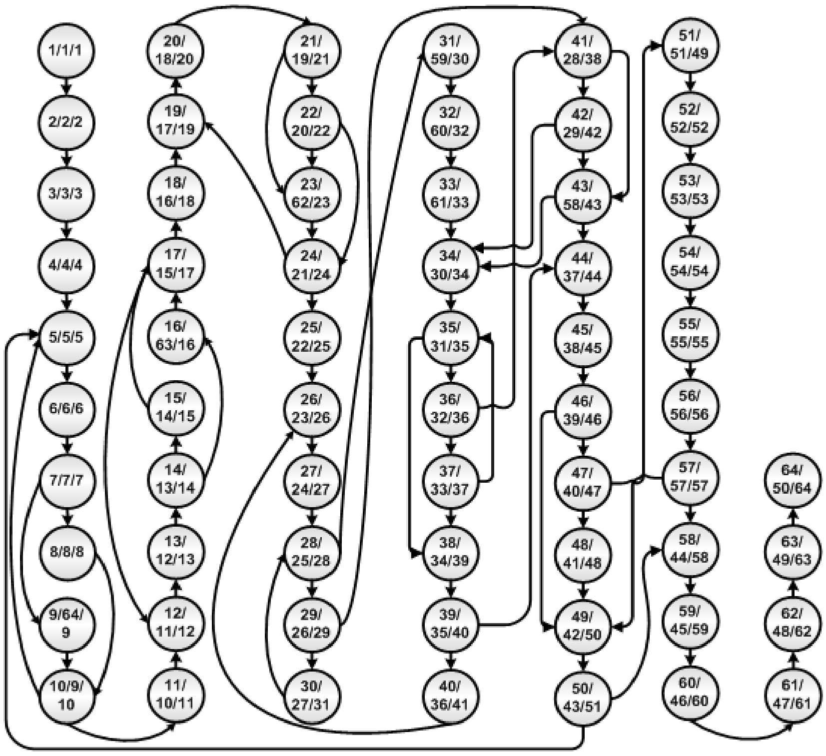

Рис. 2. Управляющий граф алгоритма обнаружения пачек двоично-квантованных сигналов.

	1/1/1	2/2/2	3/3/3	4/4/4	5/5/5	6/6/6	7/7/7	…	64/50/64
1/1/1	0	1	1	1	1	1	1	…	1
2/2/2	0	0	1	1	1	1	1	…	1
3/3/3	0	0	0	1	1	1	1	…	1
4/4/4	0	0	0	0	1	1	1	…	1
5/5/5	0	0	0	0	1	1	1	…	1
6/6/6	0	0	0	0	1	1	1	…	1
7/7/7	0	0	0	0	1	1	1	…	1
…	…	…	…	…	…	…	…	…	…
58/44/58	0	0	0	0	0	0	0	…	1
59/45/59	0	0	0	0	0	0	0	…	1
60/46/60	0	0	0	0	0	0	0	…	1
61/47/61	0	0	0	0	0	0	0	…	1
62/48/62	0	0	0	0	0	0	0	…	1
63/49/63	0	0	0	0	0	0	0	…	1
64/50/64	0	0	0	0	0	0	0	…	0

Определяя зоны управляющего графа G, получим в результате 8 зон.

Зона 1 содержит вершины 5, 6, 7, 8, 10. Зона 2 содержит вершины 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 41, 42, 34, 35, 36, 37, 38, 39, 40. Зона 3 содержит вершины 12, 13, 14, 15, 17. Зона 4 содержит вершины 19, 20, 21, 22, 24. Зона 5 содержит вершины 26, 27, 28, 29, 30, 41, 42, 34, 35, 36, 37, 38, 39, 40. Зона 6 содержит вершины 28, 29, 30.

Зона 7 содержит вершины 41, 42, 34, 35, 36. Зона 8 содержит вершины 49, 50, 51, 52, 53, 54, 55, 56, 57.

Заключение

Разработан новый алгоритм, который является более простым в программной реализации и имеет трудоемкость не более $O(n^3)$. Предлагаемый алгоритм может быть использован при анализе программных и технических систем, работа которых представлена в виде управляющих графов.

ЛИТЕРАТУРА

1. *Евстигнеев В.А.* Применение графов в программировании / под ред. А.П. Ершова. – М.: Наука. Гл. ред. физ.-мат. науки, 1985.
2. *Касьянов В.Н., Евстигнеев В.А.* Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003.
3. *Евстигнеев В.А.* Локальный алгоритм отыскания бикомпонент в ориентированном графе // Журнал вычислительной математики и математической физики. – 1978. – Т. 18, № 5. – С.1345-1349.
4. *Карзанов А.В.* Экономный алгоритм нахождения бикомпонент // Труды 3-й Зимней школы по математическому программированию и смежным вопросам. – 1970. – Вып. 2. – С.343-347.
5. *Касьянов В.Н.* Об одном алгоритме выделения бикомпонент в ориентированном графе // Системное и теоретическое программирование. – Новосибирск, 1974. – С.235-243.
6. *Лейфман Л.Я.* Эффективный алгоритм разбиения ориентированного графа на бикомпоненты // Кибернетика. – 1966. –№ 5. – С.19-23.
7. *Фараджев И.А.* Алгоритм выделения бикомпонент ориентированного графа // Труды 3-й Зимней школы по математическому программированию и смежным вопросам. – 1970. – Вып. 3. – С.650-654.
8. *Tarjan R.E.* Depth-first search and linear graph algorithm // SIAM J. Computing. – 1972. – Vol. 1. – P.215-225.
9. *Кузьмин С.З.* Основы теории цифровой обработки радиолокационной информации. – М.: Сов. радио, 1974.

Статья представлена к публикации членом редколлегии Е.А. Шеленком.

E-mail:
Гусеница Ярослав Николаевич – yaromir226@mail.ru.