панель элементов схемы, меню с функциональными возможностями (печать анализа результатов и выполнение расчётов).

Модуль построения схемы электроснабжения – отвечает за формирование связей между элементами сети на программном уровне в виде, который будет понятен для остальных модулей.

Модуль расчёта токов КЗ – позволяет осуществлять расчёт величин токов КЗ на основе характеристик элементов сети и их взаимного расположения.

Модуль анализа результатов и формирования статистики – предназначен для фиксирования величин токов КЗ при различных внешних условиях (расположению элементов в сети электроснабжения, нагрузок) и формирования статистики в различных форматах. К основным форматам можно отнести файлы типа xls (MS Excel), txt.

База данных – позволяет хранить справочную информацию, необходимую для расчётов. Данные, хранимые в ней, используются остальными модулями.

Управляющие инструкции – это реакция программных модулей на действия пользователя.

Описанная архитектура является обобщённой, поэтому выбор языка программирования не ограничен. По архитектуре можно спроектировать программное средство, которое решит задачу расчёта токов КЗ в системе электроснабжения сельского района и будет проста в использовании.

Литература

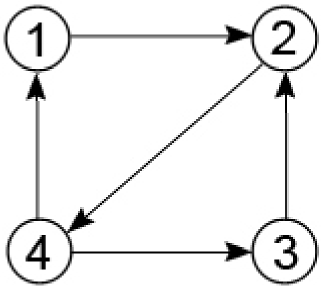
1. Программа Project StudioCS Электрика [Электронный ресурс]: Электрон. текстовые дан. – режим доступа: http://pro-spo.ru/-cad-cam-windows/1316-project-studiocs-437.

2. Программы для энергетика [Электронный ресурс]: Электрон. текстовые дан. – режим доступа: energosoft.info/soft_electrotex_121_130.html.

К.Д. Бондарь (УО «ГГУ имени Ф. Скорины», Гомель)
Науч. рук. **Е.А. Дей**, канд. физ.-мат. наук, доцент

КОМПЬЮТЕРНАЯ ОБРАБОТКА ГРАФОВ ПРИ РЕШЕНИИ ПРИКЛАДНЫХ ЗАДАЧ

Графом называется структура данных, представляющая собой совокупность множества вершин и множества ребер [1–3]. Граф называется взвешенным, если его ребру поставлено в соответствие некоторое числовое значение (весовой коэффициент). В ориентированном графе для каждого ребра указаны начальная и конечная вершины (рисунок 1).



	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	1	0	0
4	1	0	1	0

Рисунок 1 – Пример ориентированного графа и его матрицы смежности

Теория графов является одним из основных инструментов при решении таких задач как расчет свойств многоатомных молекул, проектирование печатных плат, планирование транспортных маршрутов. Например, схему электрической цепи, содержащей резисторы, конденсаторы, катушки индуктивности и источники напряжения, можно рассматривать как взвешенный ориентированный граф, в котором каждому ребру сопоставлены две переменные (напряжение и сила тока), необходимые для выполнения законов Кирхгофа и определенных физических соотношений [1]. Преимущества графов заключаются в их наглядности и естественной аналогии с реальным миром, со структурой описываемых объектов.

Основным способом описания графа в памяти компьютера является матрица смежности − квадратная матрица размером $N*N$, где N − число вершин в графе. Для ориентированного графа элемент матрицы $M[i,\ j]$ равен единице, если направленное ребро (дуга) выходит из вершины i и заканчивается в вершине j, в ином случае на пересечении i-й строки и j-го столбца будет стоять ноль (рисунок 1). Такое описание соединений в графе позволяет быстро определять, связаны вершины ребром или нет. Во взвешенном графе элемент матрицы, определяющий ребро из вершины i в вершину j, равен не единице, а величине весового коэффициента [1].

В докладе рассмотрены два алгоритма нахождения кратчайшего расстояния между вершинами взвешенного графа: алгоритм Флойда-Уоршолла [1,2] и алгоритм Форда-Беллмана [3]. Для них составлены программы на языке Java и выполнено практическое исследование быстродействия при обработке модельных графов с различным числом вершин.

Алгоритм Флойда-Уоршолла для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа был разработан в 1962 году. На каждом шаге алгоритм рассчитывает рабочую матрицу, которая содержит длины кратчайших путей между всеми вершинами графа. В начале расчета рабочая матрица заполняется длинами рёбер графа, а если ребра нет, то бесконечно большой длиной. Далее выполняются три вложенных цикла, в ходе которых обновляется ответ для каждой пары вершин i и j способом перебора промежуточной вершины k. В итоге будет найден минимальный вес результирующего ребра [1,2].

Работа алгоритма была протестирована на трех различных матрицах смежности, отличающиеся количеством вершин графа, записанных в текстовых файлах. Это были матрицы 10x10, 25x25 и 50x50. Время обработки тестовых матриц по данному алгоритму составило 4, 12 и 38 миллисекунд соответственно (рисунок 2).

Весовые коэффициенты для матрицы смежности размером 10x10 имеет вид:

№	1	2	3	4	5	6	7	8	9	10
1	0	20	0	0	15	0	0	40	0	0
2	20	0	25	10	0	0	0	0	0	0
3	0	25	0	28	0	20	0	0	30	0
4	0	10	28	0	15	24	22	0	0	0
5	15	0	0	15	0	0	0	15	0	0
6	0	0	20	24	0	0	9	0	0	8
7	0	0	0	22	0	9	0	35	0	0
8	40	0	0	0	15	0	35	0	0	0
9	0	0	30	0	0	0	0	0	0	16
10	0	0	36	0	0	8	0	0	16	0

Матрица кротчайших расстояний между вершинами имеет вид:

№	1	2	3	4	5	6	7	8	9	10
1	0	20	45	30	15	54	52	30	75	62
2	20	0	25	10	25	34	32	40	55	42
3	45	25	0	28	43	20	29	58	30	28
4	30	10	28	0	15	24	22	30	48	32
5	15	25	43	15	0	39	37	15	63	47
6	54	34	20	24	39	0	9	44	24	8
7	52	32	29	22	37	9	0	35	33	17
8	30	40	58	30	15	44	35	0	68	52
9	75	55	30	48	63	24	33	68	0	16
10	62	42	28	32	47	8	17	52	16	0

Программа выполнилась 4 миллисекунд

Рисунок 2 – Результат обработки графа с 10 вершинами по алгоритму Флойда-Уоршолла

Алгоритм Форда-Беллмана находит кратчайшие пути от одной вершины графа до всех остальных с предварительным поиском минимального веса среди всех вершин. Алгоритм был предложен в работе Форда 1956 года и развит в работе Беллмана в 1958 году [3]. Реализован в 1969 году как основной алгоритм маршрутизации для сети ARPANET.

Алгоритм обрабатывает взвешенный ориентированный граф из вершин и дуг. В графе могут быть как петли, так и дуги отрицательного веса, однако нет циклов отрицательного веса и между любой парой вершин не может быть более одной дуги в одном направлении. Требуется найти расстояние от первой вершины до всех остальных.

Для каждого из тестовых графов время выполнения программы составило 0.7, 2.3 и 8.6 секунд соответственно (рисунок 3).

```
Введите номер вершины, от которой необходимо найти кратчайшие расстояния
до всех остальных вершин, если такие насстояния имеются
5
Кратчайшее расстояние от вершины №5 до вершины №1 равно 15
Кратчайшее расстояние от вершины №5 до вершины №2 равно 25
Кратчайшее расстояние от вершины №5 до вершины №3 равно 43
Кратчайшее расстояние от вершины №5 до вершины №4 равно 15
Кратчайшее расстояние от вершины №5 до вершины №6 равно 39
Кратчайшее расстояние от вершины №5 до вершины №7 равно 37
Кратчайшее расстояние от вершины №5 до вершины №8 равно 15
Кратчайшее расстояние от вершины №5 до вершины №9 равно 63
Кратчайшее расстояние от вершины №5 до вершины №10 равно 47
Программа выполнялась 692 миллисекунд
```

Рисунок 3 – Результат обработки графа с 10 вершинами по алгоритму Беллмана-Форда

Тестовые расчеты показывают, что алгоритм Беллмана-Форда работает на несколько порядков медленнее. При этом он дает возможность поиска кратчайшего пути из выбранной вершины, в то время как алгоритм Флойда-Уоршолла выдает все кратчайшие расстояния между всеми вершинами.

Составленные программы будут использованы при решении прикладных задач по обработке графов.

Литература

1. Свами, М. Графы, сети и алгоритмы. / М. Свами, К. Тхуласираман. – М. : Мир, 1984. – 455 с.

2. Андерсон, Дж. Дискретная математика и комбинаторика / Дж. Андерсон. – М. : Изд. дом «Вильямс», 2004. – 960 с.

3. Асанов, М.О. Дискретная математика: графы, матроиды, алгоритмы. – Изд. 2-е. / М.О. Асанов, В.А. Баранский, В.В. Расин. – СПб. : Изд-во «Лань», 2010. – 368 с.