Повреждение блока по любой причине является событием, противоположным по отношению к неповреждению блока, поэтому вероятность повреждения блока

$$q_{бл} = 1 - 0,9692298 = 0,0307702$$

Таким образом вероятность поломки блока составляет 0,0307702.

Из всего выше перечисленного можно сделать вывод, что элементы теории вероятности применяются во всех сферах жизни человека, а в частности в энергетике. Она помогает при проектировке и эксплуатации различных объектов системы. Теория вероятности является основой для дисциплины «теория надежности» которую изучают на энергетических кафедрах.

Литература

1. Студенческий портал Studfiles: [Электронный ресурс]. URL: https://studfile.net/preview/7798227/#2. (Дата обращения: 14.04.2023).

УДК 519.177

ПОСТРОЕНИЕ МАРШРУТОВ РАЗЛИЧНОЙ ДЛИНЫ В ГРАФЕ ПО МАТРИЦЕ СМЕЖНОСТИ

Ханяк Е.Д.
Научный руководитель – Юринок В.И., к.т.н., доцент

Целью научной работы является анализ инфраструктуры городского транспорта и определение всех возможных путей выбранной длины, количество которых будет определяться при помощи возведения в степень матрицы смежности.

Пусть дан ориентированный невзвешенный граф G с n вершинами, и число ребер l. Требуется для каждой пары вершин i и j найти количество путей, состоящих ровно из l ребер и представить в виде маршрутов. При этом в пути могут повторяться вершины сколько угодно раз. Граф задан матрицей смежности $M(G)$ и изображен на рис. 1.

$$M(G) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

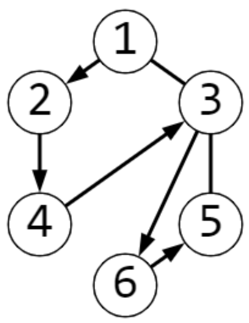

Рис. 1. Визуальное представление рассматриваемого графа

Очевидно, что данная матрица показывает количество путей для длины $l = 1$. Представим, что требуется определить маршруты, где число ребер l=2. В таком случае, для определения количества путей из i в j необходима такая вершина, которая будет смежной для i и при этом j будет смежной вершиной для данной. Обозначим через A_l найденную матрицу ответов, A_{l+1} — матрица ответов, которую надо найти. Тогда будет верна формула:

$$(a_{l+1})_{ij} = \sum_{k=1}^{n} (a_l)_{ik} \cdot m_{kj}$$

Из данной формулы можно увидеть, что это произведение матриц, т.е.:

$$A_{l+1} = A_l \cdot M \,;\, A_l = \underbrace{M \cdot M \cdot \ldots \cdot M}_{l \quad раз} = M^l$$

Для более эффективного нахождения количества путей длины 2 в программе реализован алгоритм бинарного возведения в степень (рис. 2), реализованный при помощи языка программирования C++.

```
vector <vector <int>> multiplication(vector <vector <int>> a, vector <vector <int>> b) {
    vector <vector <int>> c(n, vector <int>(n));
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            c[i][j] = 0;
            for (int k = 0; k < n; k++)
                c[i][j] += a[i][k] * b[k][j];
        }

    return c;
}


vector <vector <int>> bin_power(vector <vector <int>> a, int x) {
    if (x == 1) {
        return a;
    } else if (x % 2 == 0) {
        vector <vector <int>> b = bin_power(a, x / 2);
        return multiplication(b, b);
    } else {
        vector <vector <int>> b = bin_power(a, x - 1);
        return multiplication(a, b);
    }
}
```

Рис. 2. Фрагмент программы (бинарное возведение в степень)

Для вывода всех путей с количеством ребер l в программе реализован алгоритм поиска в глубину с настраиваемой глубиной и «массивом предков».

Поиск глубину — один из способов обхода графа. Его суть заключается в том, чтобы идти «вглубь» графа пока это возможно. Как только идти становится невозможно, алгоритм возвращается к ближайшей вершине с инцидентным неиспользованным ребром, и продолжает путь уже через это ребро.

Пусть входными данными для программы будет граф G (рис. 1) и количество ребер $l = 2$. Тогда выходными данными будут матрица А и список всех путей, подходящих под условие входных данных.

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Общее количество путей $1+1+1+1+1+1+2+1+1+1+1+1+1+1+1=16$. Каждый из них изображен на рис. 3.

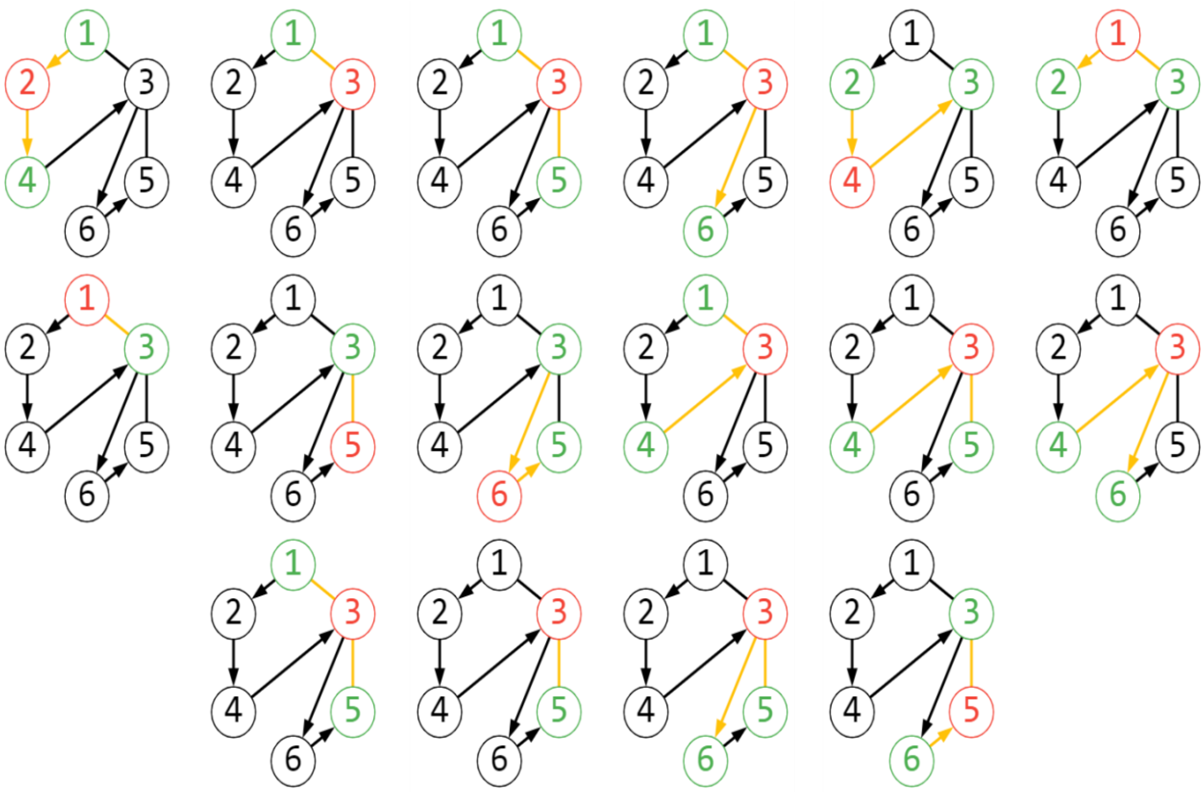

Рис. 3. Все возможные маршруты с количеством ребер $l = 2$

Так, например, на первом графе на Рис. 3 изображен схема с маршрутом 1→2→4, на втором 1→3→1 и т.д.

Программа также способна работать с более длинными маршрутами (3, 4 и т.д.).

Таким образом, удалось реализовать построение маршрутов различной длины в графе по матрице смежности, при помощи программы, написанной на языке программирования C++, в которой реализованы поиск в глубину и бинарное возведение матрицы в степень.

Практическая ценность исследований заключается в возможности применения математической модели в реальных задачах, где требуется многократное движение по выбранным точкам.

УДК 12.345.67

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММ АППРОКСИМАЦИИ ФУНКЦИЙ В СИСТЕМЕ MATHCAD

Мордвинцев К.С.
Научный руководитель – Рудый А.Н., канд.физ.-мат.н., доцент

В работе проводится анализ различных алгоритмов аппроксимации данных на примере конечного потребления природного газа как топливно-энергетического ресурса.

При измерении каких-либо величин иногда возникает проблема в определении функции, по которой они распределены, а также их коэффициенты. Так же с помощью аппроксимации данных появляется возможность спрогнозировать дальнейшее поведение измеряемой величины. В качестве исходных данных рассмотрим конечное потребление природного газа (включая попутный). Данные приведены в Таблице 1.

Таблица 1 – Конечное потребление природного газа.

Год	2014	2015	2016	2017	2018	2019	2020
Объем, м^3	4257	3845	3992	4033	4060	3914	3738

Для аппроксимации воспользуемся несколькими методами в среде Mathcad. Изначально предположим, что данная зависимость является квадратной.

Метод обратной матрицы.

При вычислении производных от суммы квадратов отклонений искомой функции относительно начальных данных получается система уравнений. Для решения данной системы используется метод обратной матрицы: