Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине: «Параллельные методы и алгоритмы»

Студент	Макаров Тимофей Геннадьевич
Группа	РК6-22М
Тип задания	Лабораторная работа
Тема	Спектральный метод балансировки нагрузки МВС

Студент ____________________ **_Макаров Т.Г.**

подпись, дата *фамилия, и.о.*

Преподаватель ____________________ **_Карпенко А.П.**

подпись, дата *фамилия, и.о.*

Оценка ____

Москва, 2023 г.

Оглавление

Цель лабораторной работы

Цель выполнения лабораторной работы – изучение статической балансировки загрузки многопроцессорной вычислительной системы (МВС) спектральным методом.

Постановка задачи балансировки нагрузки

Задачу балансировки загрузки МВС можно поставить в терминах теории графов: разрезать граф (Q, D) на N (по числу процессоров в системе) непересекающихся подграфов так, чтобы суммы весов узлов в каждом из подграфов были приблизительно равны, а сумма весов ребер, которые инцидентны узлам из разных подграфов, была минимальна. Известно несколько алгоритмов приближенного решения этой задачи, наиболее известным из которых является иерархический графовый алгоритм.

Постановка задачи бисекции графа

Рассмотрим простейший вариант спектрального алгоритма бисекции графа, когда вычислительные сложности всех процессов $Q_i, i \in [1:n]$ одинаковы и величина n четна. Поставим задачу разделения графа (Q, D) на два подграфа таким образом, чтобы суммарные вычислительные сложности подграфов были равны (т.е. были равны количества процессов в каждом из подграфов), а количество разрезанных ребер было минимально. Поставленная задача является двухкритериальной задачей. Поэтому бисекция графа (Q, D), найденная с помощью данного алгоритма, не является оптимальной ни по одному из ранее заданных критериев.

Схема спектрального алгоритма бисекции графа

Определим матрицу смежности $\boldsymbol{A}$ размера $n \times n$ для графа (Q, D), такую что:

$$a_{ij} = \begin{cases} 0, \text{если вершины } i \text{ и } j \text{ не связаны ребром;} \\ 1, \text{если вершины } i \text{ и } j \text{ связаны ребром.} \end{cases}$$

Определим матрицу $\boldsymbol{B} = diag(b_i), i \in [1{:}n]$ степеней вершин графа (Q, D), где b_i равняется числу рёбер, инцидентных i-ой вершине.

Также введём матрицу Лапласа

$$\boldsymbol{L} = \boldsymbol{B} - \boldsymbol{A}.$$

Известно, что собственные значения $\lambda_i, i \in [1{:}n]$ матрицы Лапласа являются вещественными. Найдём эти значения и соответствующие им нормализованные собственные векторы $U_i,\ i \in [1{:}n]$. Упорядочим собственные числа в порядке возрастания так, что $\lambda_1 \le\ \lambda_2 \le \cdots \le\ \lambda_n$. Также упорядочим собственные векторы в порядке возрастания соответствующих собственных значений.

Найдём среднее значение $\bar{u}$ компонентов $u_{2,i}, i \in [1{:}n]$ вектора U_2 – нормализованного собственного вектора, соответствующего второму собственному числу λ_2 . Вершины графа (Q, D) , которым соответствуют значения $u_{2,i} < \bar{u}$ отнесём к первому подграфу, а оставшиеся вершины – ко второму. В случае, если несколько величин $u_{2,i}$ имеют значение, равное $\bar{u}$, распределяем соответствующие вершины между подграфами равномерно.

Реализация алгоритма

Граф (Q, D) варианта 5, полученный из графа-шаблона, изображен на рисунке 1.

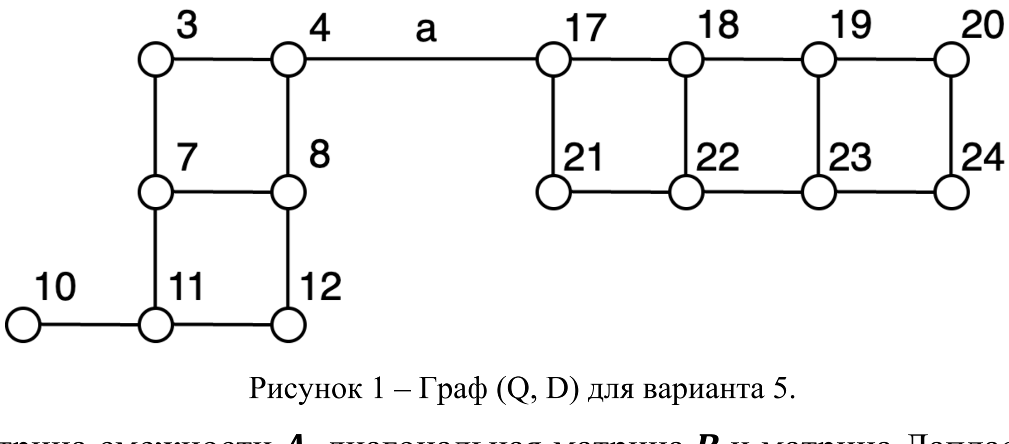

Рисунок 1 – Граф (Q, D) для варианта 5.

Матрица смежности $\boldsymbol{A}$, диагональная матрица $\boldsymbol{B}$ и матрица Лапласа для данного графа имеют следующий вид:

$$
\boldsymbol{A} = \begin{bmatrix}
0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0
\end{bmatrix}
$$

$$
\boldsymbol{B} = \begin{bmatrix}
\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{2}
\end{bmatrix}
$$

$$
\boldsymbol{L} = \begin{bmatrix}
\mathbf{2} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & \mathbf{3} & 0 & \mathbf{-1} & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & 0 & \mathbf{3} & \mathbf{-1} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & \mathbf{-1} & \mathbf{3} & 0 & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{-1} & 0 & \mathbf{-1} & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & \mathbf{3} & \mathbf{-1} & 0 & 0 & \mathbf{-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 & 0 & \mathbf{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 & 0 & \mathbf{-1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{2} & 0 & 0 & 0 & \mathbf{-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & 0 & \mathbf{2} & \mathbf{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{-1} & \mathbf{2}
\end{bmatrix}
$$

Для поиска собственных значений и соответствующих собственных векторов был использован язык программирования Python, библиотека Numpy и интегрированная среда разработки PyCharm. Код программы представлен в листинге 1.

Листинг 1. Код программы, выполняющей вычисление матрицы смежности A, диагональной матрицы B, матрицы Лапласа и её собственных чисел и векторов.

```
import numpy as np
import numpy.linalg as lin

def get_incidence_matrix():
    f = open("matrix.csv")
    dim = int(f.readline())
    m = np.zeros((dim, dim))
    for line in f.readlines():
        split = line.split(' ')
        a = int(split[0]) - 1
        b = int(split[1]) - 1
        m[a][b] = 1
        m[b][a] = 1
    f.close()
    return m

def get_vertices_degrees_matrix(incidence_matrix: np.array):
    m = np.zeros(len(incidence_matrix))
    for i in range(len(incidence_matrix)):
        m[i] = np.sum(incidence_matrix[i])
    return np.diag(m)

def get_eigen(laplace: np.array):
    values, vectors = lin.eigh(laplace)
    indexes = np.argsort(values)
    values = values[indexes]
    vectors = vectors[indexes]
    vectors = np.array([vectors[i]/lin.norm(vectors[i]) for i in
range(len(vectors))])
    return values, vectors

def main():
    A = get_incidence_matrix()
    B = get_vertices_degrees_matrix(A)
    L = B - A
    eigen_values, eigen_vectors = get_eigen(L)
    np.set_printoptions(precision=4, suppress=True)
    print("Собственные числа:\n", eigen_values)
    print("Собственные векторы:\n", eigen_vectors)
    a = []
    b = []
    t = np.array([eigen_vectors[i][1] for i in range(len(eigen_vectors))])
    avg = np.average(t)
    for i in range(len(t)):
        if t[i] < avg:
            a.append(i + 1)
        if t[i] > avg:
```

```
            b.append(i + 1)
    print(a)
    print(b)

if __name__ == '__main__':
    main()
```

Собственные числа матрицы Лапласа для графа, приведённого на рисунке 1 равны 0.0000, 0.1135, 0.5981, 0.8410, 1.4247, 2.0000, 2.0698, 2.4384, 3.0000, 3.0000, 3.4419, 3.8190, 4.4353, 5.2640, 5.5542. Соответствующие собственные векторы представлены на рисунке 2.

−0.2582	−0.2123	−0.2685	0.3139	0.5166	0	0.2356	0.1693	−0.0300	0.5013	−0.0845	−0.1762	−0.0442	−0.2402	0.1285
−0.2582	−0.1302	−0.2990	0.1514	0.2135	0	−0.2751	−0.4107	0.1876	0.0292	0.0829	0.4047	0.3561	0.3231	−0.2760
−0.2582	−0.2704	−0.0774	0.2124	0.0837	0	0.2587	0.3365	−0.1576	−0.5306	0.0390	−0.0843	−0.2486	0.4610	−0.1807
−0.2582	−0.2431	−0.1322	0.2263	−0.2359	0	−0.0951	−0.2123	0.4052	−0.4429	−0.1701	−0.2267	0.0398	−0.4551	0.2155
−0.2582	−0.3667	0.5344	−0.5136	0.3506	0	−0.0936	−0.1613	0.1876	0.0292	−0.0972	−0.1674	−0.1051	0.0817	−0.0258
−0.2582	−0.3251	0.2148	−0.0817	−0.1489	0	0.1001	0.2320	−0.3751	−0.0584	0.2374	0.4719	0.3612	−0.3485	0.1175
−0.2582	−0.3012	0.0589	0.1248	−0.6688	0	−0.0720	−0.0450	−0.0300	0.5013	−0.0467	−0.1348	−0.1647	0.2462	−0.0937
−0.2582	0.0798	−0.3175	−0.2134	0.0556	0	−0.3964	−0.1877	−0.3751	−0.0584	0.2180	0.0713	−0.5068	−0.0361	0.3610
−0.2582	0.2002	−0.1372	−0.1961	−0.0366	0.5000	−0.0128	−0.0670	−0.3751	−0.0584	−0.3720	−0.2199	0.1592	−0.1858	−0.43500
−0.2582	0.2756	0.1281	0.0588	−0.0064	0.5000	−0.0176	0.1256	0.1876	0.0292	0.4423	−0.2625	0.2878	0.2398	0.3612
−0.2582	0.3133	0.3177	0.2912	0.0692	0	−0.4711	0.4292	0.1876	0.0292	−0.2219	0.2757	−0.2147	−0.1070	−0.1402
−0.2582	0.1603	−0.3264	−0.4160	−0.0892	−0.500	−0.0808	0.3724	0.1876	0.0292	0.1928	−0.2433	0.2120	−0.0554	−0.2110
−0.2582	0.2225	−0.1401	−0.2688	−0.1070	0	0.4020	0.0244	0.1876	0.0292	−0.4959	0.3712	−0.0095	0.2171	0.3889
−0.2582	0.2819	0.1271	0.0318	−0.0426	0	0.4676	−0.2916	0.1876	0.0292	0.3984	0.1591	−0.3576	−0.2501	−0.3474
−0.2582	0.3155	0.3173	0.2787	0.0462	−0.500	0.0504	−0.3138	−0.3751	−0.0584	−0.1224	−0.2391	0.235	0.1094	0.1372

Рисунок 2 – собственные векторы матрицы Лапласа.

Таким образом, компоненты вектора U_2 равны -0.2123, -0.1302, -0.2704, -0.2431, -0.3667, -0.3251, -0.3012, 0.0798, 0.2002, 0.2756, 0.3133, 0.1603, 0.2225, 0.2819, 0.3155.

В результате получаем бисекцию графа (Q, D), представленную на рисунке 3.

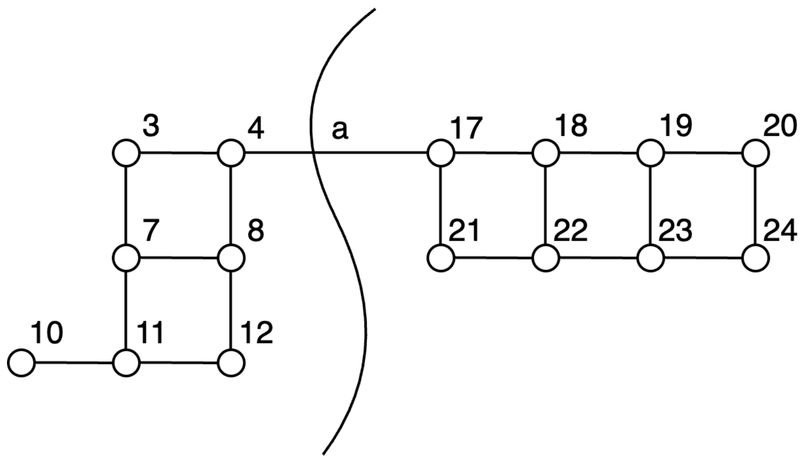

Рисунок 3 – Бисекция графа (Q, D), полученная с помощью спектрального алгоритма.

Заключение

В процессе выполнения лабораторной работы была проведена статическая балансировка загрузки многопроцессорной вычислительной системы, был изучен и реализован спектральный метод бисекции графа как приближенного способа решения задачи отображения вычислительных процессов на архитектуру многопроцессорной ЭВМ. Получено оптимальное разбиение графа на две части в соответствии с вариантом.

Список использованных источников

1. Карпенко А.П., Федорук Е.В. Параллельные вычисления. Учебнометодическое пособие к лабораторным работам по курсу «Параллельные вычисления». М.: НУК ИУ МГТУ им. Н.Э. Баумана, 2010. 72 с.