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 «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.04 Программная инженерия»

ОТЧЕТ

по практикуму №2 по обработке графов знаний на вычислительной платформе «Тераграф»

Название: Обработка графов знаний на вычислительной платформе «Тераграф»

Дисциплина: Архитектура ЭВМ

Студент	ИУ7-53Б		Н. Д. Лысцев
	Группа	Подпись, дата	И. О. Фамилия
Преподаватель			А. Ю. Попов
		Подпись, дата	И. О. Фамилия

Москва, 2023 г.

Цели работы

В ходе практикума ознакомиться с архитектурой и принципами работы вычислительного комплекса Терапраф, выполнить практические задания по программированию гетерогенных ядер обработки графов, ознакомиться с библиотеками для обработки и визуализации графов. Доступ к вычислительному комплексу осуществляется с использование облачной платформы Терапраф Cloud, обеспечивающей одновременный доступ многих пользователей к гетерогенным ядрам обработки, входящим в состав микропроцессора Леонард Эйлер.

На основе изложенных сведений необходимо разработать распределенное приложение обработки и визуализации графов, функционирующее в системе Терапраф.

Практикум состоит из трех этапов:

- Исследование принципов функционирования вычислительного комплекса Терапраф
- Практикум по программированию гетерогенного вычислительного узла
- Командный практикум по генерации музыкальных композиций на основе графов знаний

Практикум №2. Обработка и визуализация графов в вычислительном комплексе Тераграф

Практикум посвящен освоению принципов представления графов и их обработке с помощью вычислительного комплекса Тераграф. В ходе практикума необходимо ознакомиться с вариантами представления графов в виде объединения структур языка C/C++, изучить и применить на практике примеры решения некоторых задач на графах. По индивидуальному варианту необходимо разработать программу хост-подсистемы и программного ядра sw_kernel, выполняющего обработку и визуализацию графов.

Конвейер визуализации графов

Визуализация графа — это графическое представление вершин и ребер графа. Визуализация строится на основе исходного графа, но направлена на получение дополнительных атрибутов вершин и ребер: размера, цвета, координат вершин, толщины и геометрии ребер. Помимо этого, в задачи визуализации входит определение масштаба представления визуализации. Для различных по своей природе графов, могут быть более применимы различные варианты визуализации. Таким образом задачи, входящие в последовательность подготовки графа к визуализации, формулируются исходя из эстетических и эвристических критериев. Графы можно визуализировать, используя:

- 2D графическую сцену - наиболее часто применяемый случай, обладающий приемлемой вычислительной сложностью (порядка $O(n^2 \log n)$);
- 3D графическую сцену - такой вариант позволяет выполнять перемещение камеры наблюдения, что увеличивает возможное количество визуализируемых вершин;
- Иерархическое представление - граф представляется в виде иерархически вложенных подграфов (уровней), что позволяет более наглядно представить тесно связанные компоненты первоначального графа.

Для визуализации графов было определено несколько показателей качества, позволяющие получить количественные оценки эстетики и удобства графического представления. Алгоритмы раскладки, в большинстве случаев, нацелены на оптимизацию следующих показателей:

- Меньшее количество пересечений ребер: выравнивание вершин и ребер для получения наименьшего количества пересечений ребер делает визуализацию более понятной и менее запутанной.
- Минимум наложений вершин и рёбер.
- Распределение вершин и/или рёбер равномерно.
- Более тесное расположение смежных вершин.

• Формирование сообществ вершин из сильно связанных групп вершин.

Для больших графов метод выделения сообществ является предпочтительным, так как позволяет выявить скрытые кластеры вершин, показать центральную вершину сообщества, минимизировать количество внешних связей между сообществами.

Таким образом визуализация графов представляет собой многостадийный алгоритмический процесс. Кратко стадии процесса визуализации представлены на следующем рисунке.

Рисунок 1 – Процесс визуализации графа

Кратко поясним представленный процесс:

• Исходный граф может быть представлен различными способами, повышающими эффективность алгоритмов их обработки. Такой граф служит исходными данными для задачи визуализации.

• На первом этапе формируется граф визуализации, содержащий для каждой вершины и ребра дополнительные атрибуты, значение которых и требуется определить в ходе этого процесса. Могут быть использованы дополнительные атрибуты, позволяющие выявить свойства вершин и более наглядно визуализировать структуру графа. Частым случаем является определение свойства центральности для вершин и ребер. В конечном итоге, для каждой вершины требуется хранить еще и ее координаты (x,y), цвет (color), радиус окружности для представления на сцене визуализации (size).

• Далее выполняется ряд алгоритмов для определения свойств вершин и ребер. В практикуме используется алгоритм Дейкстры для поиска кратчайших путей, на основе которого рассчитывается центральность вершин.

• На следующем этапе происходит выделение групп вершин, входящих в так называемые сообществ: связность вершин внутри сообщества превосходит связность за его пределами. Примеры алгоритмов поиска сообществ представлены в примерах.

• Для каждого сообщества определяется область экрана для его размещения (алгоритмы глобального размещения, global placement).

• Далее выполняется размещение вершин сообщества внутри каждой выделенной области. На данном этапе применяются алгоритмы, позволяющие представить связи небольшой группы вершин. Частым случаем является применение алгоритмов, основанных на имитации сил притяжения и отталкивания. сформированных на основе информации о вершинах и ребрах.

• На заключительном этапе происходит передача готового графа визуализации в библиотеку. осуществляющую отрисовку сцены визуализации. В практикуме используется библиотека bokeh.

Индивидуальное задание

Выбрать пять музыкальных произведений различных композиторов и жанров. Произведение должно быть доступно в формате midi. Используя код Проекта 6 получить по две визуализации для каждого музыкального произведения.

Для выполнения данного задания я выбрал 5 музыкальных произведений:

- Кино -- Группа крови
- Кино -- Кукушка
- Заглавная тема сериала «Doctor Who»
- Eminem -- MockingBird
- Pirates of the Caribbean - Main Theme

Визуализация проводилась на основе модулярности Ньюмана и алгоритма Фрухтерамана-Рейнгольда.

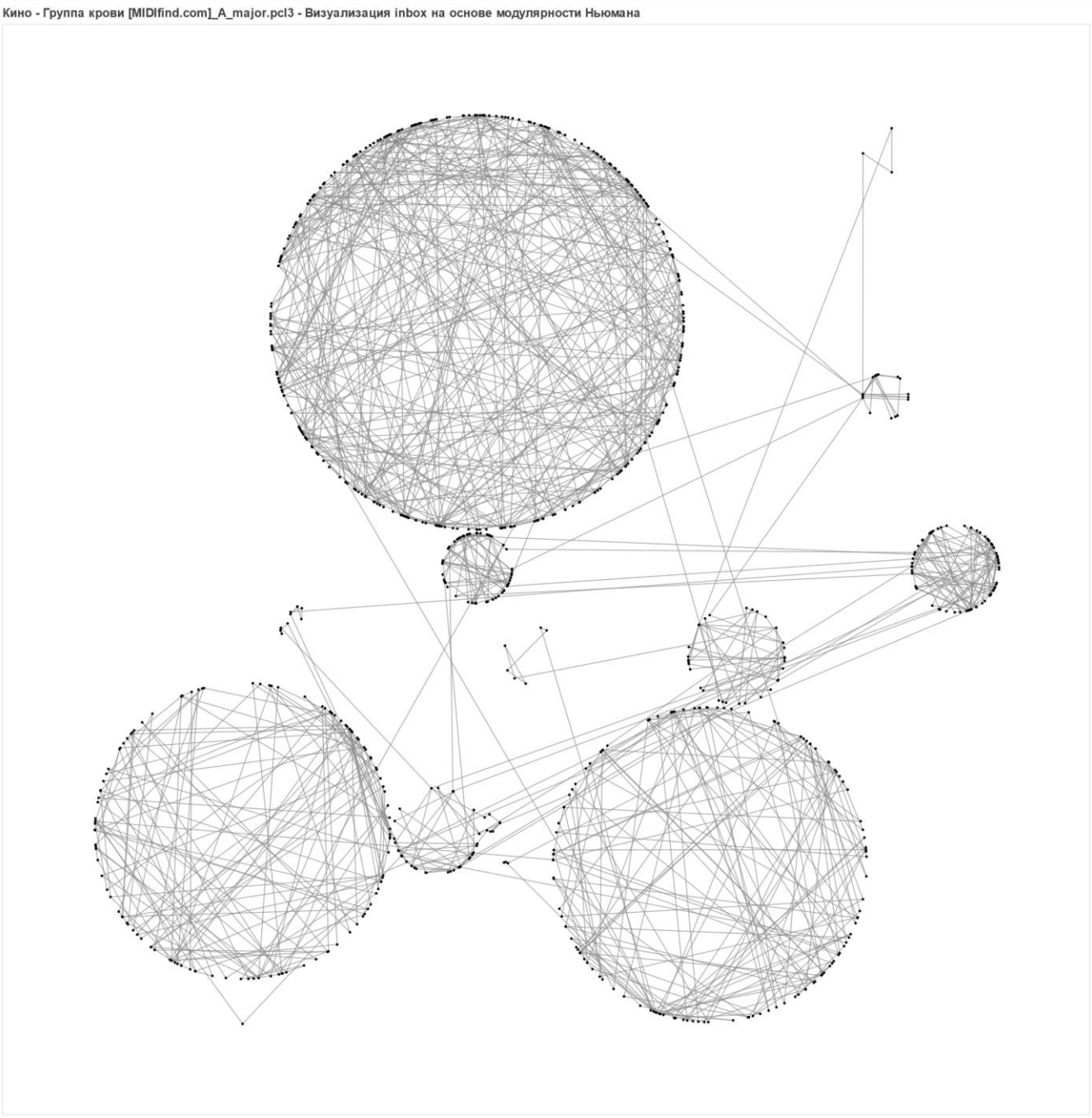

Рисунок 1. Кино -- Группа крови. Модулярность Ньюмана

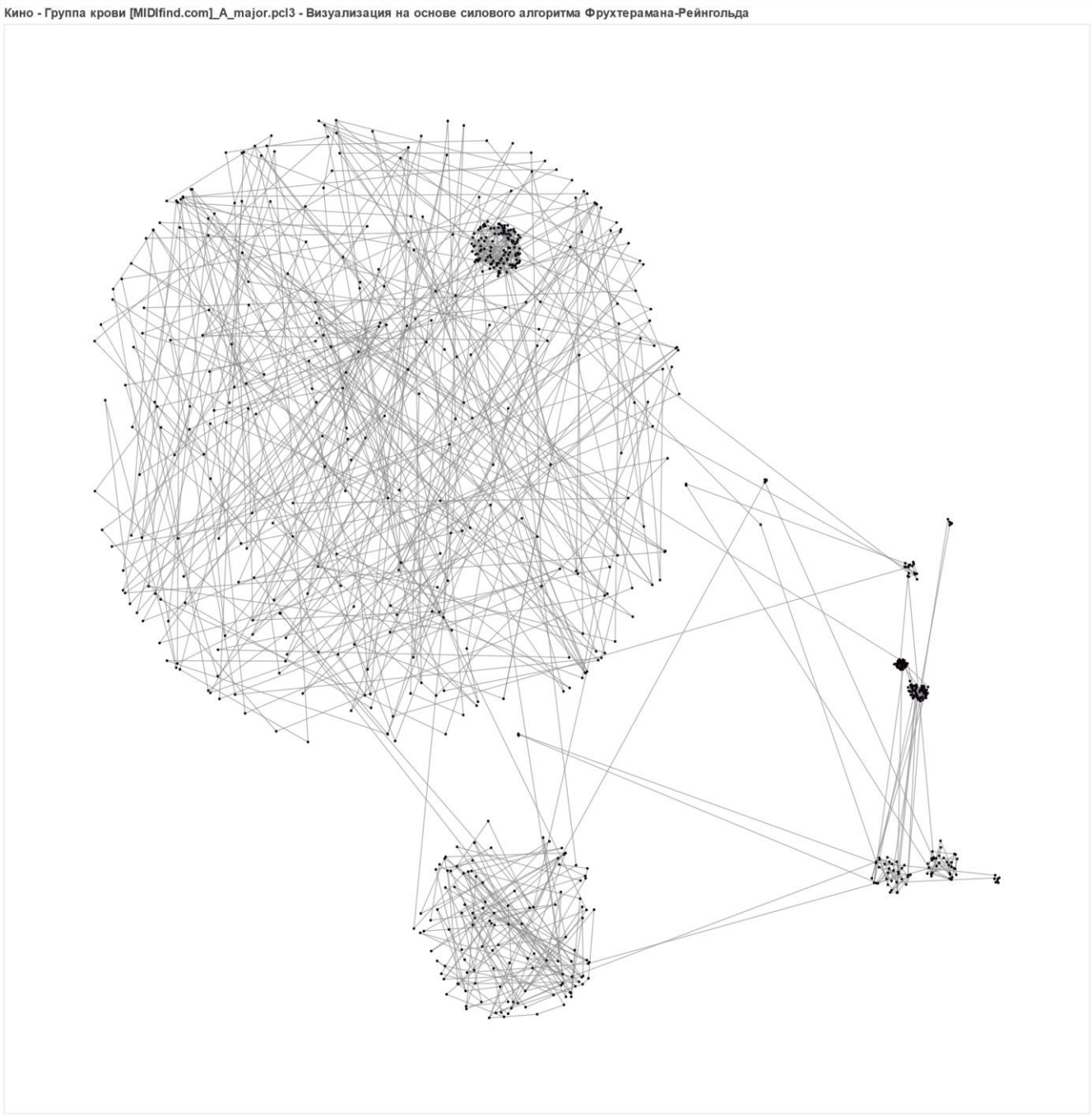

Рисунок 2. Кино -- Группа крови. Силовой алгоритм Фрухтурамана-Рейнгольда

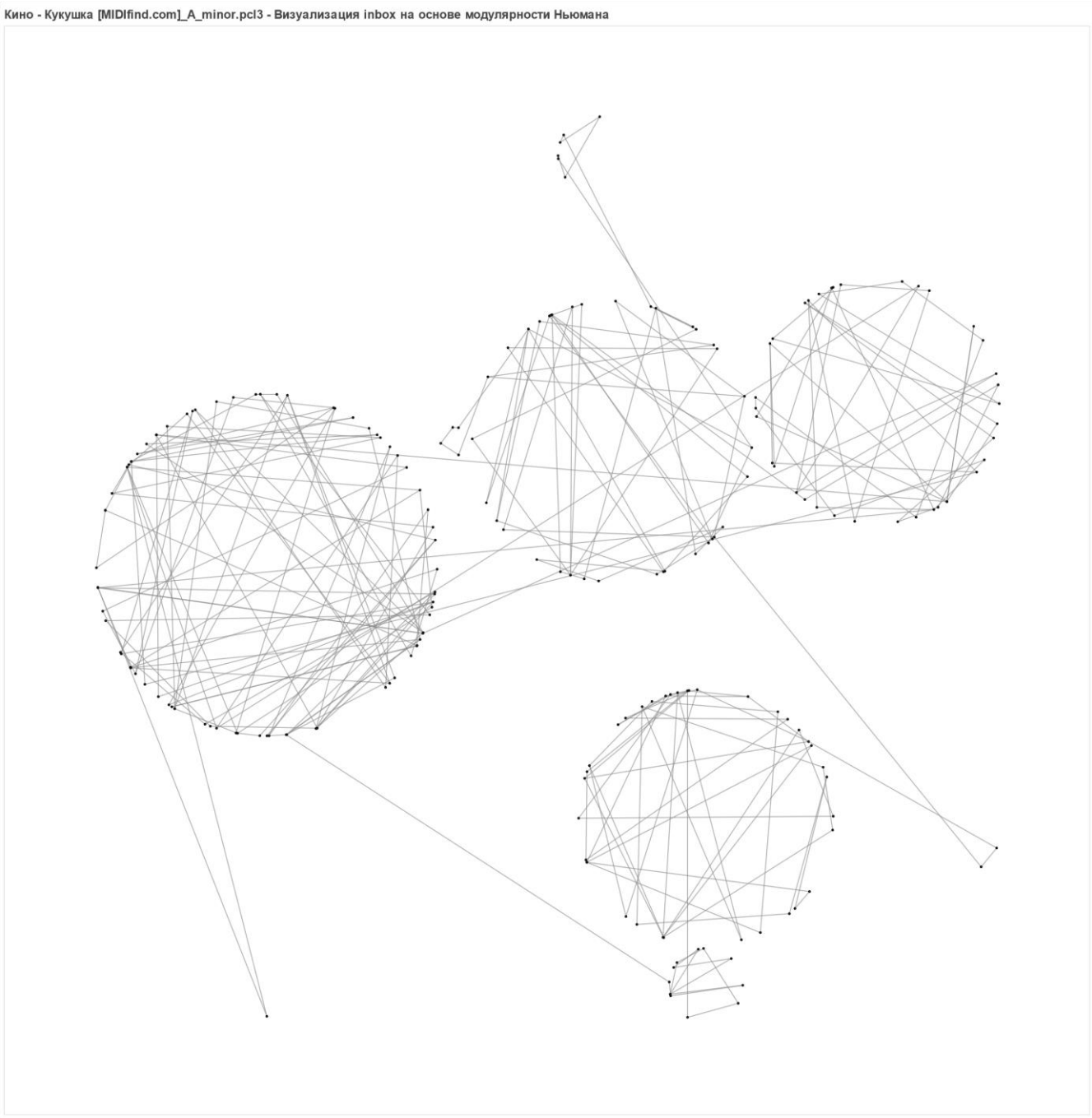

Рисунок 3. Кино -- Кукушка. Модулярность Ньюмана

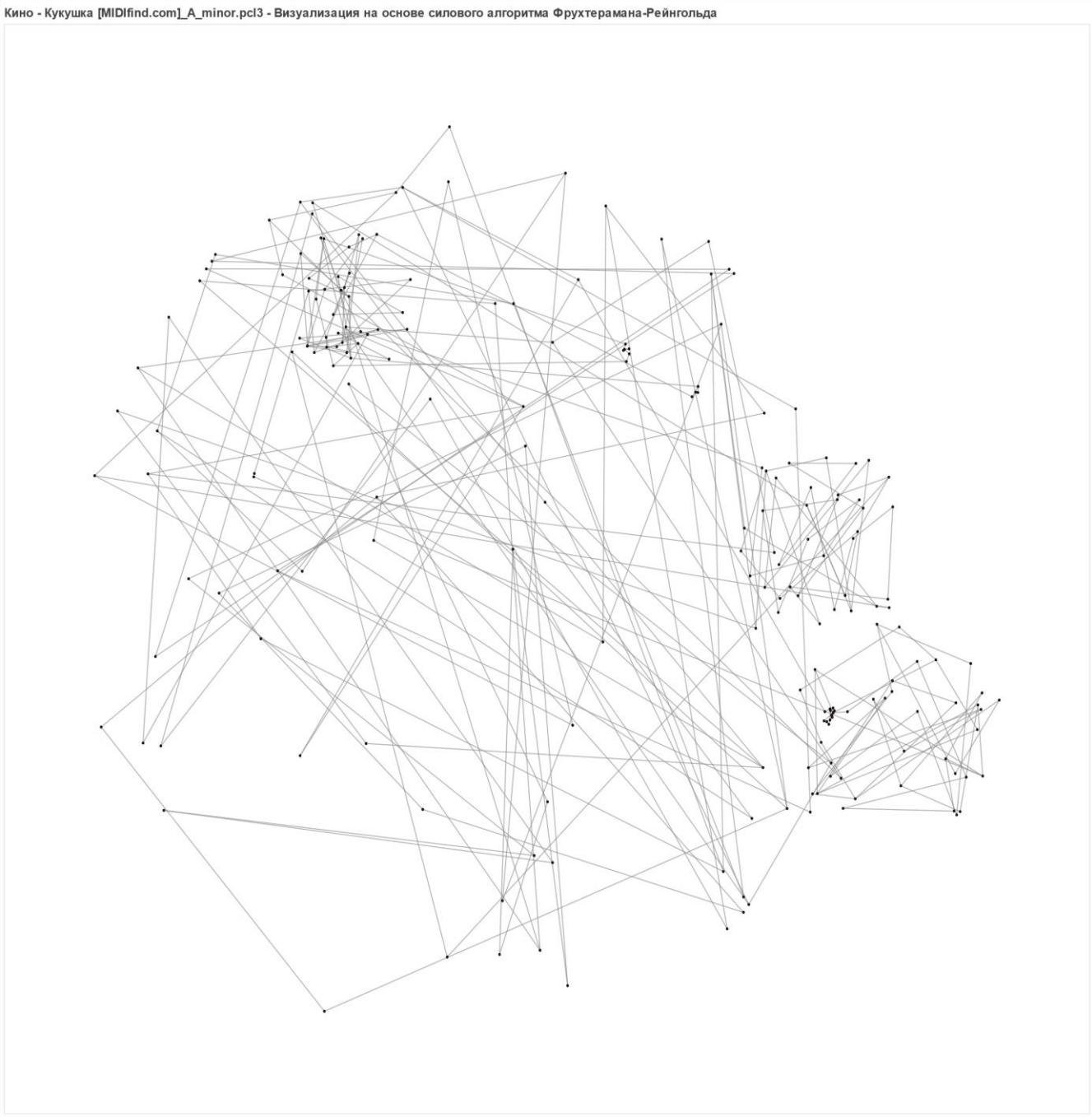

Рисунок 4. Кино -- Кукушка. Силовой алгоритм Фрухтерамана-Рейнгольда

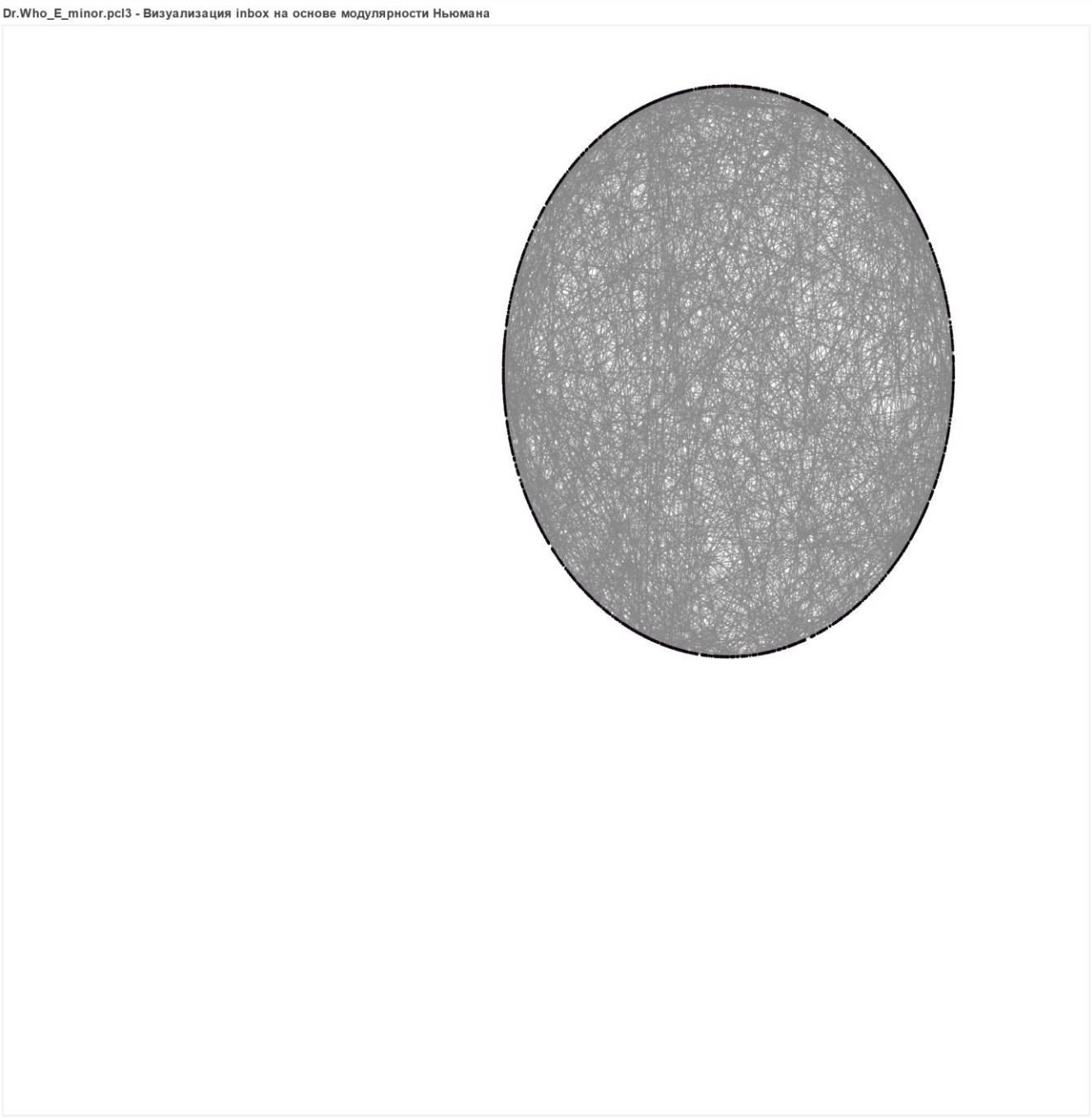

Рисунок 5. Заглавная тема сериала "Doctor Who". Модулярность Ньюмана

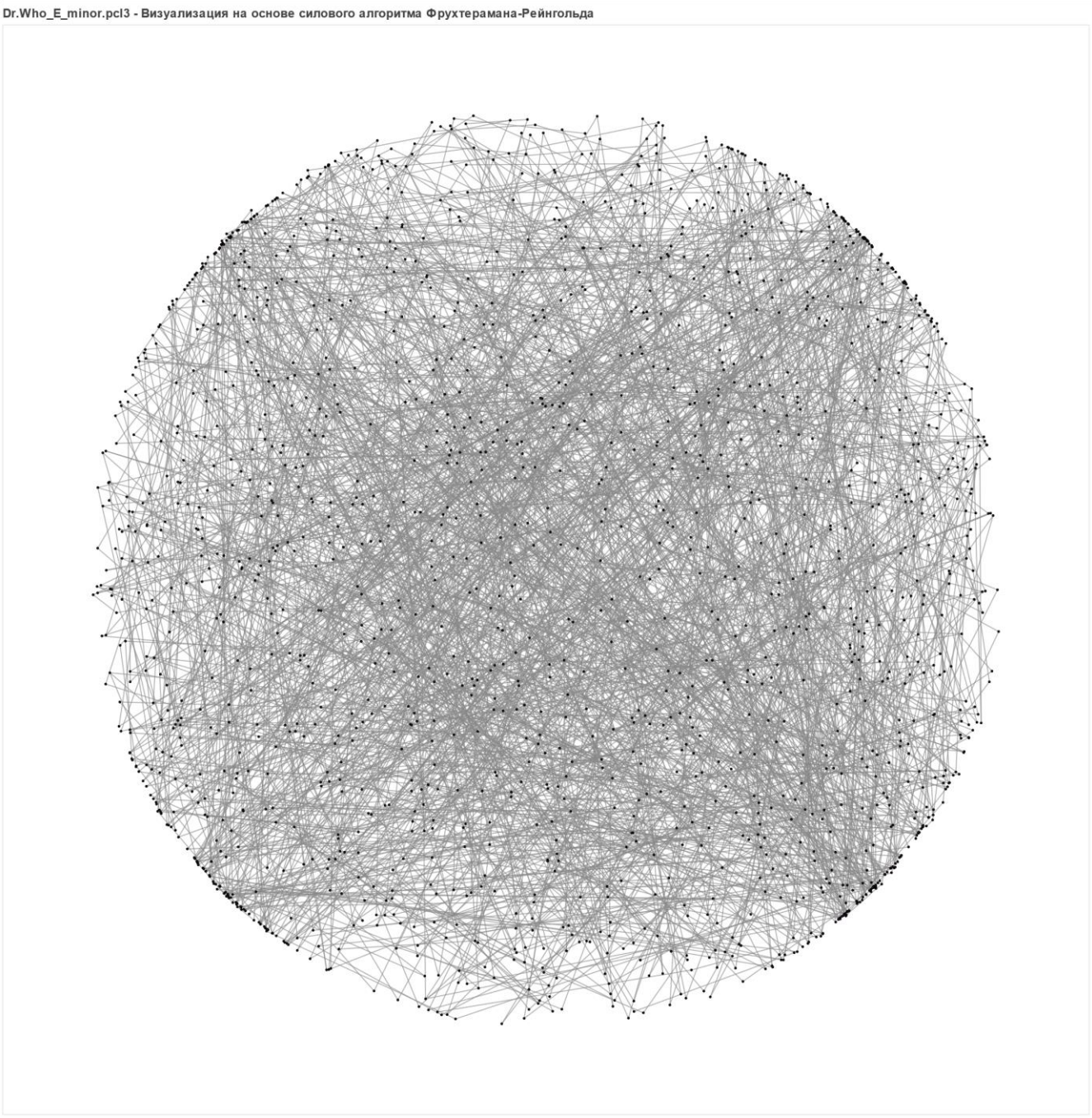

Рисунок 6. Заглавная тема сериала "Doctor Who". Силовой алгоритм Фрухтерамана-Рейнгольда

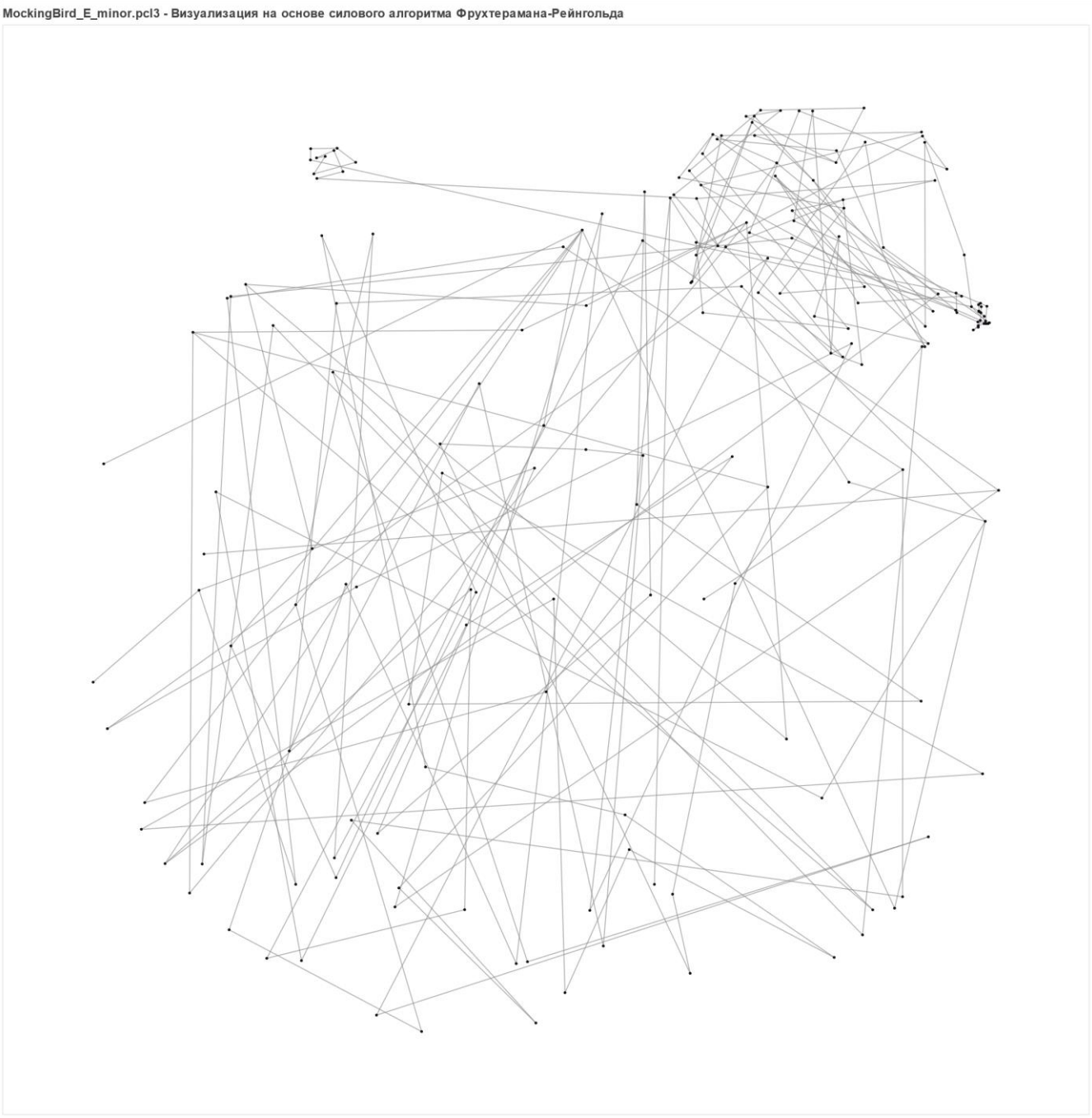

Рисунок 7. Eminem -- MockingBird. Силовой алгоритм Фрухтерамана-Рейнгольда

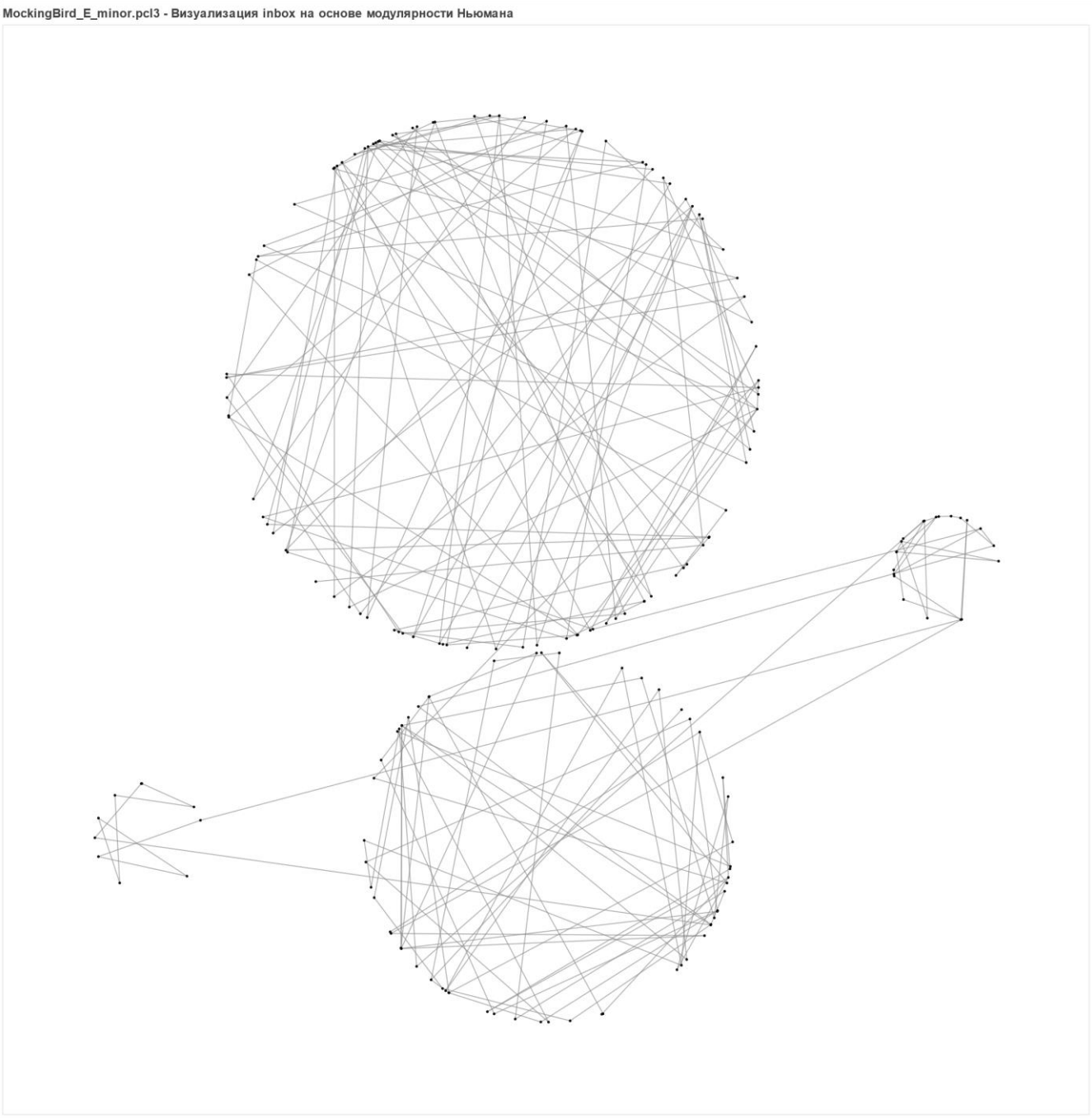

Рисунок 8. Eminem -- MockingBird. Модулярность Ньюмана

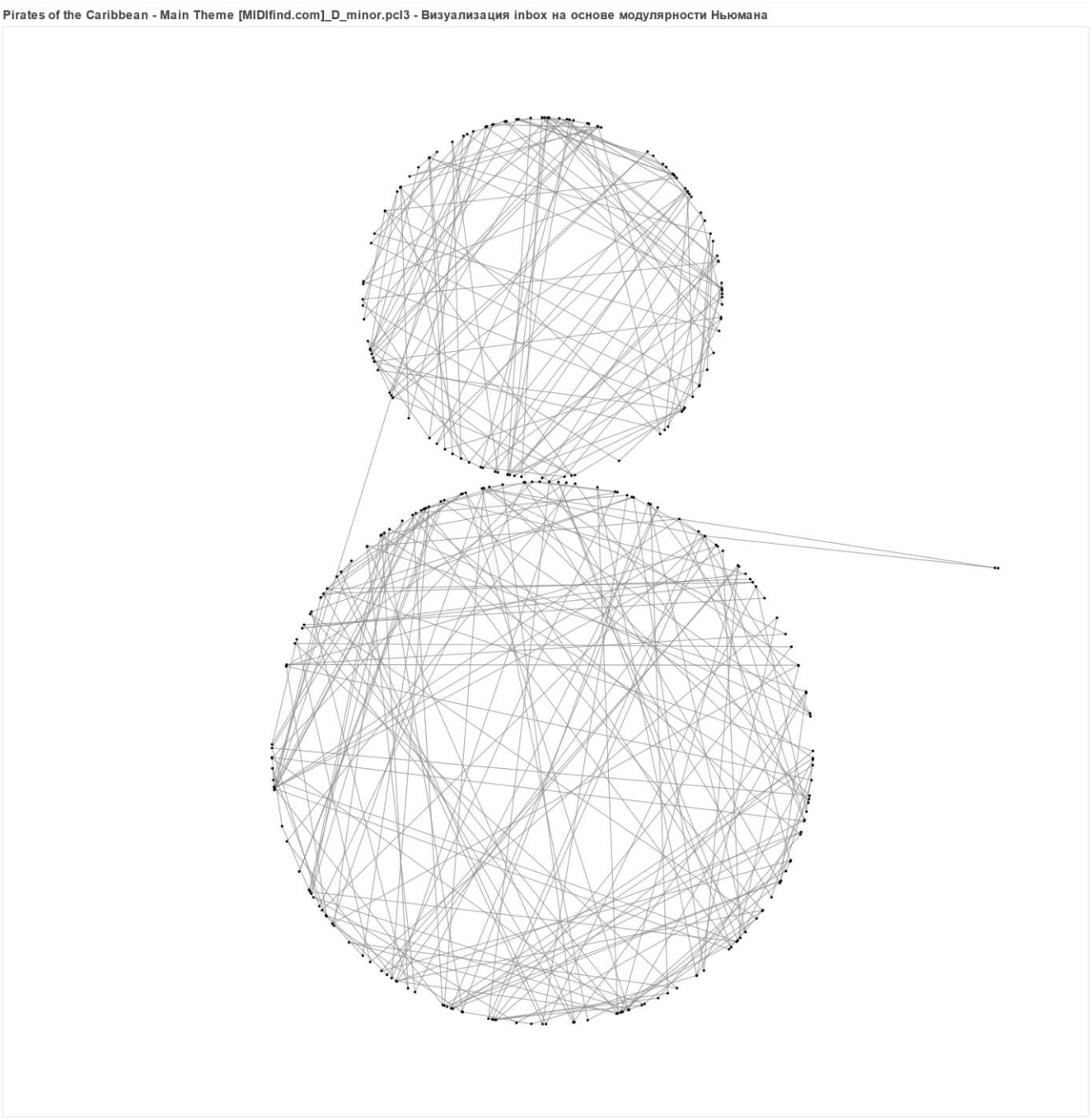

Рисунок 9. Pirates of the Caribbean - Main Theme. Модулярность Ньюмана

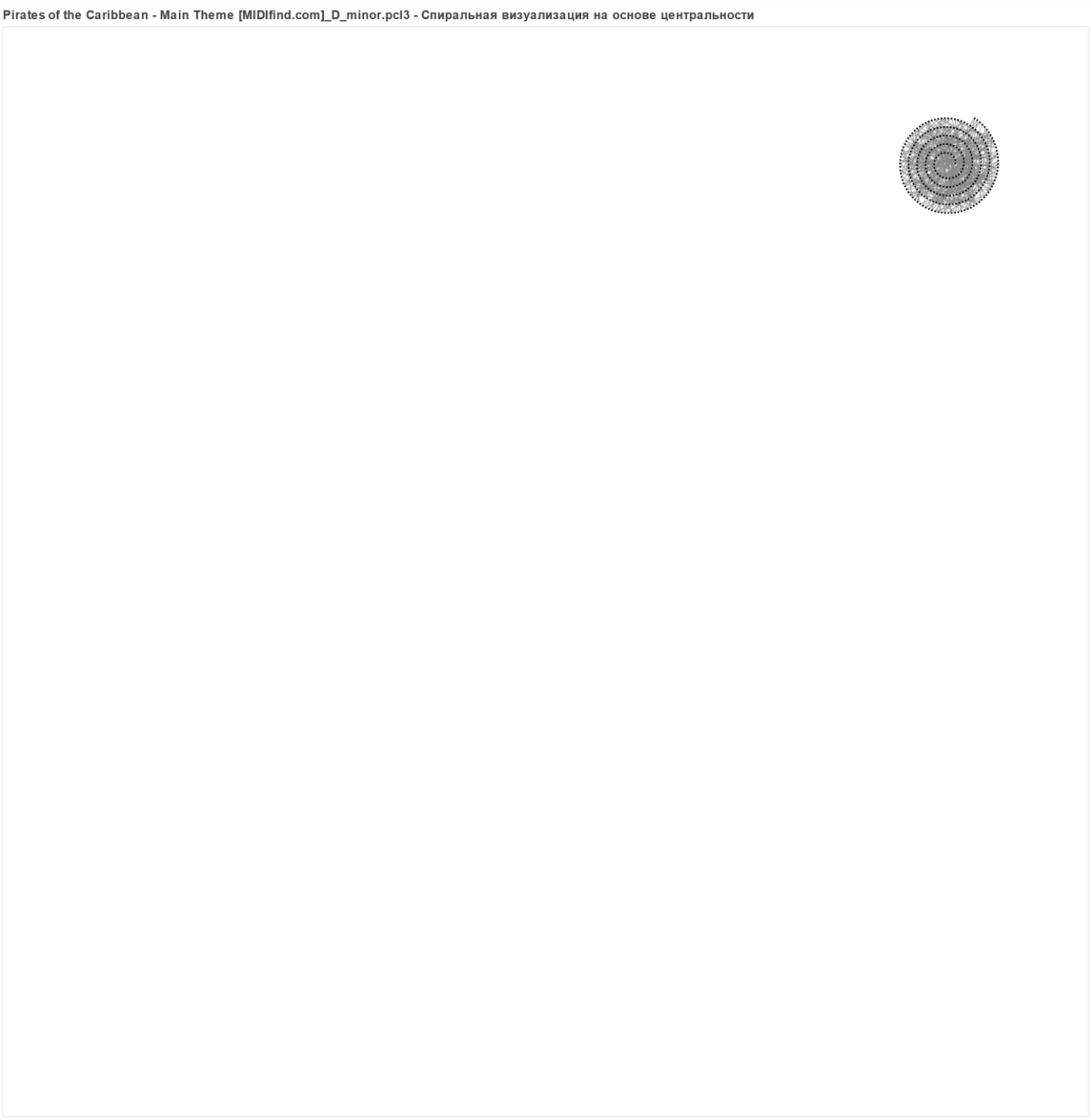

Рисунок 10. Pirates of the Caribbean - Main Theme. Спиральная визуализация на основе центральности