

На Рис.4 можно видеть движение цилиндра вдоль оси Ox в любом положении в сфере.

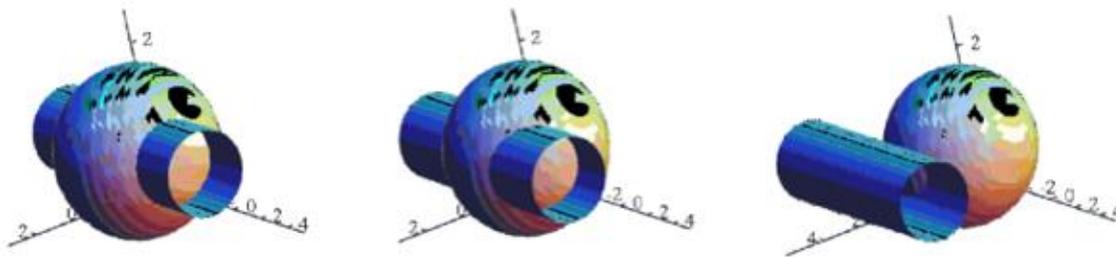


Рис.4. Движение цилиндра

Также можно представить движение цилиндра вдоль оси Oy и Oz .

Данная работа в первую очередь позволяет представить пересечение сложных поверхностей в учебных целях: для нахождения объёма тел, статических моментов, моментов инерции и в других инженерных задачах. Также, чтобы проиллюстрировать физические законы, биологические процессы или исторические события. В медицине трехмерные модели движения могут использоваться для визуализации внутренних органов.

УДК 519.174.7

АЛГОРИТМ ВЕРШИННОЙ И РЁБЕРНОЙ РАСКРАСКИ ПЛАНАРНЫХ ГРАФОВ

Корнейчик М. С.

Научный руководитель - Юринок В. И., к.т.н., доцент

В данной работе описан алгоритм вершинной и рёберной раскраски планарных графов. Решение задачи раскраски графов является ключевым элементом многих алгоритмов и методов, используемых в различных областях, от компьютерных наук до планирования и оптимизации ресурсов. Эта задача позволяет эффективно моделировать и оптимизировать распределение ограниченных ресурсов с учетом различных ограничений и требований.

“Жадный” алгоритм вершинной раскраски — это простой, но мощный метод для раскраски графа. Основная идея заключается в том, чтобы пройти по каждой вершине графа и назначить ей цвет, который еще не был использован среди ее соседей.

Рассмотрим основные этапы работы алгоритма:

1. Инициализация: на этом этапе создается пустая карта цветов, где каждой вершине будет назначен цвет.

2. Сортировка вершин: далее вершины сортируются по убыванию степени, то есть сначала раскрашиваются вершины с наибольшим количеством соседей.

3. Выбор цвета: для каждой вершины выбирается минимальный цвет, который еще не был назначен ее соседям.

4. Назначение цвета: назначенный цвет добавляется в карту цветов.

5. Повторение: процесс повторяется для всех вершин графа.

6. Хроматическое число: после раскраски всех вершин, хроматическое число графа определяется как максимальное значение цвета в карте цветов плюс один.

Рассмотрим особенности работы “жадного” алгоритма вершинной раскраски.

Выбор порядка вершин: “жадный” алгоритм чувствителен к порядку, в котором вершины добавляются для раскраски. Это означает, что разные порядки могут привести к разным результатам. Один из подходов — раскрашивать вершины в порядке убывания степени, но существуют и другие стратегии, такие как раскраска по порядку возрастания степени или использование эвристических методов для определения порядка.

Хроматическое число: хроматическое число — это минимальное количество цветов, необходимое для раскраски графа таким образом, чтобы смежные вершины имели разные цвета. “Жадный” алгоритм не всегда находит это минимальное число, но дает приближенное решение.

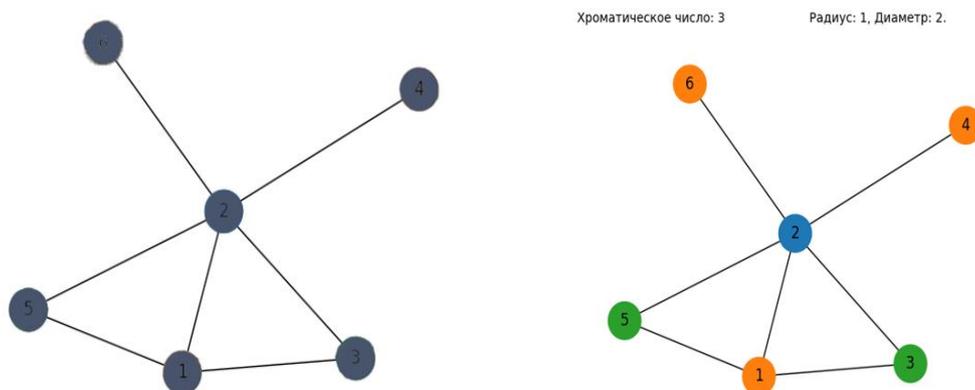


Рис. 1. Снимок экрана исходного и с раскрашенными вершинами графов

Рассмотрим каждый этап алгоритма реберной раскраски графа:

1. Выбор начального ребра: начинаем с выбора любого ребра в графе. Это ребро станет отправной точкой для раскраски.

2. Присвоение ребру цвета: присваиваем выбранному ребру цвет из доступного набора цветов. Этот шаг устанавливает начальное состояние раскраски.

3. Проверка инцидентности: для каждого следующего ребра, которое мы собираемся раскрасить, проверим, не инцидентно ли оно уже раскрашенным рёбрам. Инцидентные рёбра — это рёбра, которые имеют общую вершину.

4. Выбор цвета для нового ребра: если новое ребро инцидентно раскрашенному ребру, выбираем для него цвет, который ещё не использовался для инцидентных рёбер. Это гарантирует, что никакие два инцидентных ребра не будут иметь одинаковый цвет.

5. Повторение процесса: продолжаем процесс, раскрашивая рёбра, пока все рёбра графа не будут раскрашены.

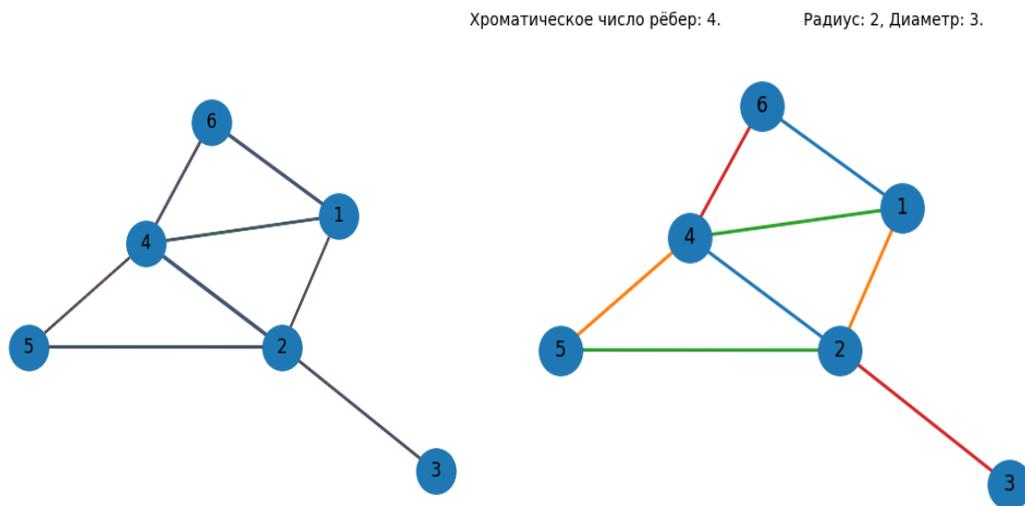


Рис. 2. Снимок экрана исходного и с раскрашенными рёбрами графов

Важно отметить, что алгоритм может быть выполнен различными способами, в зависимости от выбранного метода раскраски. Например, “жадный” алгоритм будет выбирать цвета последовательно для каждого ребра, не стремясь к оптимальности, в то время как более сложные алгоритмы могут стремиться к минимизации общего количества используемых цветов.

В заключение отметим, что решение задачи раскраски графов играет значительную роль в различных областях, подчеркивая свою важность как ключевой элемент многих алгоритмов и методов. “Жадный” алгоритм раскраски, хотя и прост в реализации, демонстрирует мощный подход к решению данной задачи.