Информация, размещенная на нашем сервисе, будет полезна как практикующим врачам, так и студентам медицинских университетов, т.к. содержит в себе хорошо проработанный, структурированный материал и имеет интуитивно понятную навигацию по основным разделам, что облегчает поиск нужной информации.

Так же облачные ресурсы нашего проекта могут быть использованы: для предоставления дополнительных вычислительных мощностей для проведения вычислительных экспериментов; как хранилище данных (хранение документов, данных по экспериментам и т.д.); для интеграции с другими сервисам в сети Интернет, для получения доступа к их потенциальным возможностям.

Список цитированных источников

1. Денисов, Д.В. Перспективы развития облачных вычислений/Д.В. Денисов // Прикладная информатика. – М.:ООО "Маркет ДС Корпорейшн", 2009. – № 5. – С. 52.

2. Уокер, Г. Основы облачных вычислений. Новый способ предоставления вычислительных ресурсов. [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.ibm.com/developerworks/ru/library/cl-cloudintro/

3. Вик, Дж. Р. Уинклер. Адаптированная выдержка из книги «SecuringtheCloud» [Электронный ресурс]. – Электронные данные. – Режим доступа: http://technet.microsoft.com/ru-ru/magazine/jj851176.aspx.

УДК 656.13.05

ГЕНЕРАЦИЯ ПЛОСКИХ КУБИЧЕСКИХ ГРАФОВ

Козеко Е.Л.
Брестский государственный технический университет, г. Брест
Научный руководитель: Шуть В.Н., к.т.н., доцент

Теория графов применяется в различных науках. Одной из важных задач данной теории, эффективное решение которой имеет большое значение, является генерация плоских кубических графов. Её результаты могут использоваться для решения различных практических задач науки, техники и управления.

Существует несколько методов генерации плоских кубических графов. В работе был предложен метод генерации с помощью установки перемычек, который был описан на ЯВУ C++. Алгоритм данного метода заключается в том, что поочередно в каждый контур исходного графа всеми возможными способами добавляется новое ребро. Таким образом, новый граф имеет n+2 вершины, где n – количество вершин исходного графа; p+3 ребер, где p – количество ребер исходного графа; g+1 граней, где g – количество граней исходного графа соответственно. Для реализации использовали матрицу инцидентности ребер и граней, которая показывает, какие ребра входят в состав граней, и позволяет определить, в какую грань будет установлена перемычка, т. е. в какие ребра будут добавляться новые вершины.

Изначально задавался плоский кубический граф, имеющий четыре вершины. Для него строилась матрица инцидентности ребер и граней, которая показывала, какие ребра входят в составы граней (если ребро p_i входило в грань g_j, в ячейке ij ставилась 1). Далее по ней заполнялась матрица контуров граней, где хранилась информация о вершинах, образующих каждую из граней. Она необходима для того, чтобы определить состав граней после добавления перемычки и составить новую таблицу инцидентности ребер и граней для описания полученного графа.

Данный метод хорош тем, что в нем нет необходимости проверять граф на планарность, что также упрощает весь алгоритм решения поставленной задачи.

После генерации всех возможных графов на одном шаге, т. е. полученных из графов с одинаковым количеством вершин, все пары графов проверяются на изоморфизм. Данная операция необходима для того, чтобы упростить и ускорить дальнейшую генерацию, т. к. удаление одного из изоморфных графов в паре позволяет уменьшить количество используемых графов. Проверка графов на изоморфизм осуществлялась в два этапа. На первом этапе проверялось необходимое, но не достаточное условие изоморфизма. Оно заключается в том, что сравниваемые графы должны иметь одинаковый состав граней, что определяется по описанной выше матрице инцидентности ребер и граней. Если данное условие не выполняется, графы не являются изоморфными. Если же выполняется, проверяется второе условие. Чтобы однозначно определить изоморфизм графов, необходимо выяснить, какие фигуры, составляющие граф, являются смежными. Для этого также можно воспользоваться матрицей инцидентности ребер и граней, где в строках единицы определяют смежность граней, т. е. если грани содержат одно и то же ребро, они являются смежными. Таким образом, если списки количества одинаковых фигур граней совпадают и каждая из фигур одного графа является смежной такому же набору фигур, как и в другом сравниваемом графе, графы являются изоморфными, следовательно, в дальнейшей генерации будет участвовать лишь один граф из рассматриваемой пары.

Заключительным этапом являлась проверка существования гамильтонова контура, что в частности определяло цель нашей работы. Для нахождения гамильтоновых циклов использовали перебор с возвратом, который имеет следующий алгоритм:

- поиск начинаем с первой вершины графа;
- пусть уже найдены первые k-1 вершин возможного цикла. Рассматриваем ребра, выходящие из последней найденной вершины, и находим ребро, соединенное с ранее не рассмотренной вершиной;
- включаем найденную вершину в цикл и помечаем ее как рассмотренную, помещая ее в множество рассмотренных вершин. Если такой вершины нет, то возвращаемся к предыдущей вершине и повторяем вычисления.

Главной целью нашей работы являлось нахождение минимального плоского кубического графа, не имеющего гамильтонов контур. Как известно, существуют такие графы с 38 вершинами (впервые был построен независимо Ледербергом Босаком и Барнеттом). Однако достичь поставленной цели не удалось. При генерации графов возникло несколько проблем, не позволяющих добиться необходимых результатов. Одна из них – большой рост графов на каждом шаге, что требует значительных вычислительных ресурсов. Время генерации увеличивается в разы, что не позволяет решить поставленную задачу. Данная проблема в большей степени связана с тем, что с ростом количества графов увеличивается время работы алгоритма проверки изоморфизма. Безусловно, время работы остальных алгоритмов также резко возрастает. Эти проблемы являются преградой для достижения цели.

Таким образом, необходимо более детально рассмотреть данную задачу. Нахождение альтернативных алгоритмов может помочь достичь желаемых результатов.

Список цитированных источников

1. Харари, Ф. Теория графов / Ф. Харари. – М: Мир, 1973. – С. 4; 88.
2. Плотников, А.Д. Дискретная математика / А.Д. Плотников. – М: Новое знание, 2005. – С. 63.