В повседневной жизни для решения многих задач часто используются случайные графы. Случайные графы нашли практическое применение во всех областях, где нужно смоделировать сложные сети - известно большое число случайных моделей графов, отражающих разнообразные типы сложных сетей в различных областях. Случайные графы применяются при моделировании и анализе биологических и социальных систем, сетей, а также при решении многих задач класса NP.

Случайные графы впервые определены венгерскими математиками Эрдёшем и Реньи в книге 1959 года «On Random Graphs»[[8]](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84#cite_note-On_Random_Graphs-8) и независимо Гильбертом в его статье «Random graphs»[[5]](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84#cite_note-Random_graphs-5).//[https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9\_%D0%B3%D1%80%D0%B0%D1%84#%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F]

Случайный граф **-** общий термин для обозначения вероятностного распределения графов. Случайные графы можно описать просто распределением вероятности или случайным процессом, создающим эти графы[[1]](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84#cite_note-Random_Graphs-1).

Наука о графах – одна из весьма красивой и богатой приложениями науке - науке о случайных графах. Эта наука находится на стыке комбинаторики, теории графов и теории вероятностей. В основе ее лежит глубокая идея о том, что мощные инструменты современной теории вероятностей должны поспособствовать более верному осознанию природы графа, призваны помочь решению многих комбинаторных и теоретико-графовых задач.

С математической точки зрения случайные графы необходимы для ответа на вопрос о свойствах *типичных* графов.

Рассмотрим одну из моделей случайного графа - модель Эрдёша – Реньи.

Модель Ердеша-Ренье

Модель Ердеша-Ренье является одной из первых моделей случайного графа. Граф построенный по этой модели представляет собой совокупность множества вершин V={1,…,n} и множества рёбер E, состоящего из рёбер полного графа , построенного на множестве V, выбранных по схеме Бернулли. Таким образом образуется случайный граф G=(V, E). Формально выражаясь, мы имеем вероятностное пространство, G(n, p)=(, , ), в котором:

Таким образом, в модели Эрдеша-Реньи каждое ребро независимо от других ребер входит в случайный граф с вероятностью p. Модель Эрдеша-Реньи на данный момент является самой изученной моделью случайных графов. [М.М. Берновский, Н.Н. Кузюрин Случайные графы, модели и генераторы безмасштабных графов]

Модель Барабаши-Альберт

Модель Барабаши-Альберт является одной из первых моделей веб-графов. Веб-граф представляет собой ориентированный мульти-граф, вершинами в котором являются какие-либо конкретные структурные единицы в Интернете: речь может идти о страницах, сайтах, хостах, владельцах и пр. Для определенности будем считать, что вершинами веб-графа служат именно сайты. А рёбрами соединяются вершины, между которыми имеются ссылки. Также Барабаши и Альберт была предложена модель предпочтительного присоединения, основная идея которой заключается в том, что при присоединении к графу новой вершины проводится некоторое количество рёбер от добавленной вершины к уже существующим, при этом вероятность появления ребра межу новой вершиной и какой-то конкретной вершиной пропорциональна степени данной вершины. Однако в своих работах Барабаши и Альберт никак не конкретизировали, какую именно из этих моделей они предлагают рассматривать. Одной из наиболее удачных и часто используемых моделей предпочтительного присоединения является модель Боллобаша-Риодана.

Модель Боллобаша-Риодана

Существуют две основных и, по сути, совпадающих модификации этой модели. В одной дается динамическое, а в другой статическое описание случайности.

Динамическая модификация

В данной модификации при добавлении n-ной вершины проводятся n новых рёбер, при этом рёбра могут быть кратными, а также петлями и даже кратными рёбрами, при создании графа с единственной вершиной проводится петля в этой точке. Таким образом вероятность появления ребра равна , где deg i –– количество уже проведенных рёбер из n в i. Очевидно, что распределение вероятностей задано корректно, поскольку

[Райгородский А.М. Модели случайных графов.]

Статическая модификация, или LCD-модель

Данная модель основывается на объекте называемом линейной хордовой диаграммой (LCD). Для построения данного объекта требуется зафиксировать на оси абсцисс 2n точек {1,…,2n}, разбить их на пары и соединить элементы каждой пары дугой, лежащей в верхней полуплоскости. Количество различных диаграмм равно

(1)

По каждой диаграмме строится граф с n вершинами и n ребрами по следующему алгоритму:

1. Идти слева направо по оси абсцисс пока не встретится правый конец какой-либо дуги, пусть позиция этой точки равна
2. Последовательность объявляется списком смежности для k-той вершины,
3. Если k<n, k увеличивается на 1, переход на шаг 1).

При построении модели LCD случайно выбирается одна из возможных LCD и вероятность каждой диаграммы равной , где – общее число диаграмм. Графы построенные по такой модели имеют те же свойства, что и графы построенные по динамической модификации схемы Боллобаша-Риодана.

***Модель Чунг-Лу***

Пусть нам задано некоторое конечное множество вершин V ={ ,… , } и степень каждой вершины , i= . Генерация графа G = (V, E) происходит следующим образом:

* Формируем множество L , состоящее из i d копий i v для каждого i от 1 до n .
* Задаем случайные паросочетания на множестве L .
* Для вершин u и v из Vs количество ребер в графе G , соединяющее их, равно числу паросочетаний между копиями u и v в L .

Сгенерированный таким образом граф соответствует степенной модели P(a,b) , описывающей графы, для которых:

[М.М. Берновский, Н.Н. Кузюрин Случайные графы, модели и генераторы безмасштабных графов]

Реализация

В ходе выполнения курсовой работы была реализована модель Барабаши-Альберт. Все расчёты производились на компьютере с intel core i5-8265U и 16 ГБ оперативной памяти. Модель реализована на Python 3.9.1 с помощью библиотек networkx, random и numpy.random.

Реализованная модель представляет собой модель растущего случайного графа предпочтительного связывания. Были реализованы две модификации: в одной из них на каждом шагу добавляется фиксированное количество рёбер, а во второй количество новых рёбер определяется распределением Пуассона.

Первая модификация

Сначала создаётся полный граф из m вершин. Затем в графе создаются n-m вершин, но рёбра ещё не проводятся. Далее создаются и инициализируются вспомогательные массивы “nodes”, “used” и “degrees”, хранящие список присоединённых к графу вершин, информацию о том использованы они или нет и степени вершин, соответственно. Затем в цикле по не присоединённым вершинам для каждой вершины, с помощью функции random.choices(), выбираются m различных вершин с которыми будет соединена новая вершина, для того чтобы не было кратных ребер создаётся массив connections, хранящий все выбранные вершины. Следующим шагом с помощью массива connections исправляются массивы used и degrees. И новая вершина добавляется в массив nodes. Случай добавления первого рассматривается отдельно: если флаг “o” поднят, добавляется грань между нулевым и первым узлами. Далее по такому же алгоритму присоединяются другие вершины.

Вторая модификация

Основное отличие второй модификации заключается в том, что на каждом шаге значение m выбирается заново по распределению Пуассона, с помощью функции numpy.random.poisson(). Остальные манипуляции производятся аналогично.

Вывод и представление данных для анализа

Также данная реализация содержит фрагмент кода, отвечающий за отображение данных о построенном графе для дальнейшего анализа. Для этого используются библиотеки matplotlib, matplotlib.pyplot и pylab. Так как графы, состоящие из 10000 элементов слишком велики для !?, то анализируются данные о распределении степеней вершин в графе. На первом шаге создаются десять графов для анализа, с помощью одной из двух описанных подпрограмм, а также создаются массивы “c” и “x” по которым будет строиться график зависимости !?. Данные о количестве вершин каждой степени записываются в массив “c”. Затем каждый элемент этого массива делится на 10, для того чтобы найти среднее значение по десяти графам. Затем, при помощи команды pylab.loglog() строится граф по полученным данным, массивы “c” и “x” сохраняются в текстовый файл.