УДК 519.7

ЭФФЕКТИВНЫЙ МЕТОД ГЕНЕРАЦИИ СЛУЧАЙНЫХ ГЕОМЕТРИЧЕСКИХ ГРАФОВ ДЛЯ МОДЕЛИРОВАНИЯ БЕСПРОВОДНЫХ СЕТЕЙ

В. В. Шахов, А. Н. Юргенсон, О. Д. Соколова

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

Ввиду высокой сложности современных сетей и стохастического характера происходящих в них процессов, основным инструментом анализа инфокоммуникационных систем является имитационное моделирование. При анализе функционирования беспроводных технологий (беспроводных сенсорных сетей, ad hoc-сетей, когнитивного радио и др.) в качестве математической модели топологии сети часто используются случайные геометрические графы, в частности UDG-графы. Следовательно, вопрос о разработке эффективного генератора таких графов является актуальным. Описан метод генерации псевдослучайных геометрических графов с наперёд заданными свойствами. Предложенный генератор превосходит существующие аналоги как по производительности, так и по качеству сгенерированных топологий.

Ключевые слова: *беспроводные сети, имитационное моделирование, топология сетей, случайные геометрические графы, генератор графов.*

DOI 10.17223/20710410/34/8

FAST METHOD FOR GENERATING RANDOM GEOMETRIC GRAPHS FOR WIRELESS NETWORKS MODELLING

V. V. Shakhov, A. N. Yurgenson, O. D. Sokolova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

E-mail: shakhov@rav.sscc.ru, nastya@rav.sscc.ru, olga@rav.sscc.ru

Due to the high complexity of modern networks as well as the stochastic nature of network processes, the simulation becomes the main tool for communication systems analysis. In simulation studies of wireless technologies (wireless sensor networks, ad hoc-networks, cognitive radio, Internet of Things etc.), the random geometric graph (RGG) is the most commonly used mathematical model of network topology. This graph is constructed by randomly placing N vertices in some metric space (according to a specified probability distribution), where each pair of vertices is connected by an edge iff the distance between them is smaller than a predetermined value (radius). Network simulation technique typically uses the rectangular region and the uniform distribution for vertices coordinates. Thus, the research and development of efficient RGG generators is an important problem. The paper describes a method for the fast generation of pseudo-random geometric graphs with the prescribed properties: 1) the vertices are uniformly distributed in the network region; 2) the region is fully covered by circles of a given radius centered at the vertices of the graph; 3) the graph is connected. To increase the generator performance, we construct an auxiliary grid covering the region. To guarantee the full coverage and graph connectivity, every cell of the grid contains at least one vertex. The corresponding cell size depends on the given transmission radius in a wireless network. The core idea of the proposed algorithm is to apply the procedure of edges construction only for nodes in adjacent cells. It allows to avoid unnecessary calculations. In the first stage of the algorithm, one vertex is randomly placed in each cell. To define the vertex coordinates, one of the standard generators of pseudo-random values for uniform distribution is used here and below. In the second part of the algorithm, the remaining vertices are randomly distributed over the region. The edges construction procedure based on adjacent cells is applied as well. The algorithm can be easily adapted for non-squared region. Experiments show that the proposed method, when compared to the traditional algorithm, drastically reduces the generation time (up to 10 times). The proposed generator is better than the existing analogues both in performance and reality of generated topologies.

Keywords: *wireless network, simulation, network topology, random geometric graph generator.*

Введение

Современные сети передачи данных можно разделить на иерархические (с централизованным управлением) и одноранговые. Обмен данными между узлами в иерархических сетях осуществляется через центральный узел. Функционирование сети с децентрализованным управлением отличается тем, что все узлы равноправны и могут взаимодействовать друг с другом без базовой станции, на расстоянии, на котором возможно произвести обмен данными. В качестве примера такой одноранговой сети можно рассмотреть беспроводную сенсорную сеть (БСС).

Беспроводная сенсорная сеть — это распределённая самоорганизующаяся сеть, состоящая из множества сенсоров, которые могут собирать данные (например, контроль температуры в заданной области) и передавать их в сток (или во внешнюю сеть) через другие сенсоры с помощью беспроводного канала. Такие сети, как правило, содержат большое количество узлов, которые имеют одинаковые радиусы передачи данных. Каждый узел может передавать информацию всем узлам, расстояние до которых не больше радиуса его передачи (рис. 1). Область покрытия подобной сети может составлять от нескольких метров до нескольких километров за счёт способности ретрансляции сообщений от одного узла к другому.

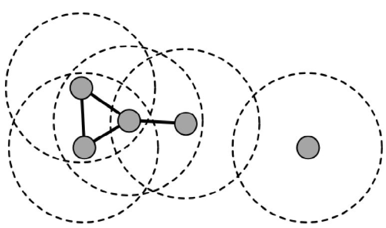


Рис. 1. Модель взаимодействия узлов в БСС

Выбор протоколов маршрутизации для сбора данных со всех узлов и передачи их в общий сток обусловлен в значительной степени спецификой расположения узлов в области. Таким образом, вопросы, связанные с топологией БСС, с выбором её оптимальной структуры, являются одними из самых важных в анализе сетей. Кроме того, для решения некоторых оптимизационных задач, например выбора эффективных параметров сенсоров, требуется проведение большого количества экспериментов, т. е. необходимо проводить имитационные исследования на различных структурах БСС. Применение имитационного моделирования позволяет в результате таких тестов совершенствовать структуру сети и оптимизировать параметры протоколов передачи данных. Для таких многочисленных тестирований необходимо генерировать большое количество моделей, адекватно представляющих топологию беспроводной сети и механизмы взаимодействия узлов. Таким образом, генерация большого количества адекватных моделей для тестирования параметров современных беспроводных сетей является актуальной задачей.

1. Определения, постановка задачи

Топологию любой сети связи можно наглядно представить как множество вершин (узлы сети) и множество рёбер (каналы связи), следовательно, при исследованиях удобно использовать графовые модели и применять методы теории графов для анализа функционирования. Наиболее часто используемыми моделями являются ориентированные графы, гиперграфы, двудольные графы, а также случайные графы [1], в которых каждое ребро существует с некоторой вероятностью.

При моделировании беспроводных технологий передачи данных также удобно использовать графовые топологии: расположение узлов моделируется точками на евклидовой плоскости, а область распространения сигнала от каждого узла (т. е. место, где сигнал может быть получен другим узлом) моделируется кругом; две точки соединяются ребром, если одна точка находится в круге, образованном другой точкой. Если все узлы имеют передатчики одинаковой мощности, то круги имеют один и тот же радиус.

При разработке имитационных моделей для анализа БСС активно используют *случайные геометрические графы* (Random Geometric Graph, RGG) [2]. Такие графы строятся путём размещения вершин случайным образом в некоторой области с заданной метрикой (обычно на плоскости), при этом каждая пара вершин соединяется ребром, если расстояние между ними не более заданной величины. Для моделирования сети обычно используется прямоугольная область, а для выбора координат вершин — равномерное распределение.

Определение 1. *Случайным геометрическим графом* называется граф $G(n, R)$, который получается в результате размещения на плоскости случайным образом n вершин, при этом две вершины соединяются ребром, если евклидово расстояние между ними не больше R.

Среди случайных геометрических графов выделяют подкласс Unit Disk Graphs, UDG-графы (графы единичных кругов). В таком графе, расположенном в евклидовой плоскости, ребро между двумя вершинами существует, если евклидово расстояние между этими вершинами не больше единицы. Использование UDG-графов для моделирования топологии беспроводных сетей передачи данных началось с работы [3].

Определение 2. Граф $G = (V, E)$ с множеством вершин V и множеством рёбер E называется *UDG-графом*, если для любых вершин u, v ребро $(u, v) \in E$ существует тогда и только тогда, когда в евклидовой метрике расстояние между u и v меньше либо равно 1.

Для UDG-графов разработано множество приближенных алгоритмов, эффективно решающих известные задачи оптимизации: поиск максимального независимого множества, раскраска графов, поиск доминирующего множества, поиск максимальной клики, разбиение графа на клики. В [4] приведены алгоритмы генерации связных случайных UDG-графов.

В большинстве публикаций [5 – 7] по тематике беспроводных сетей, в которых при анализе эффективности представленных результатов используются в качестве моделей случайные геометрические графы, опускается описание алгоритмов генерации таких графов. Для решения этой проблемы требуется аналитическое обсуждение и сравнение существующих алгоритмов генерации.

Принимая во внимание специфику беспроводных сетей, на граф, моделирующий топологию БСС, могут налагаться следующие требования [8, 9]:

— связность,
— ограничение на степень вершин,
— заданный диаметр и др.

Генерация таких псевдослучайных геометрических графов (т. е. графов с наперёд заданными свойствами) вызывает большой интерес. Традиционно при генерации графа топологии сети n узлов ($n = |V|$) распределяются случайным образом независимо друг от друга в некоторой области и некоторые вершины соединяются рёбрами по определённым правилам. Затем осуществляется проверка сети на связность и на корректность выполнения некоторых заданных условий, в частности, необходимо принимать во внимание равномерность покрытия сенсорами всей области. Например, в [10] описывается инструментарий GenSeN, способный генерировать реалистичные топологии беспроводных сенсорных сетей с такими важными характеристиками сенсорных узлов, как энергопотребление. В [11] исследуется возможность моделировать поведение сети с использованием различных протоколов и с учётом условий окружающей среды. В [12] для различных топологий сенсорных сетей сравнивается их производительность на основе таких важных показателей, как степень узла, надёжность сети, количество каналов и др. Однако, несмотря на актуальность задачи генерации реалистичных топологий беспроводных сетей, к настоящему моменту не существует стандартного алгоритма генерации случайных геометрических графов.

В данной работе рассматривается следующая задача: разработать генератор псевдослучайных геометрических графов, имеющий следующие параметры:

— n — количество вершин графа (число сетевых узлов);
— R — радиус передачи данных от одного сетевого узла к другому, т. е. если расстояние между двумя вершинами $\mathrm{dist}(v_i, v_j) \leqslant R$, то ребро между ними существует;
— область для размещения вершин является квадратом с заданной длиной стороны.

Сгенерированные графы должны обладать следующими свойствами:

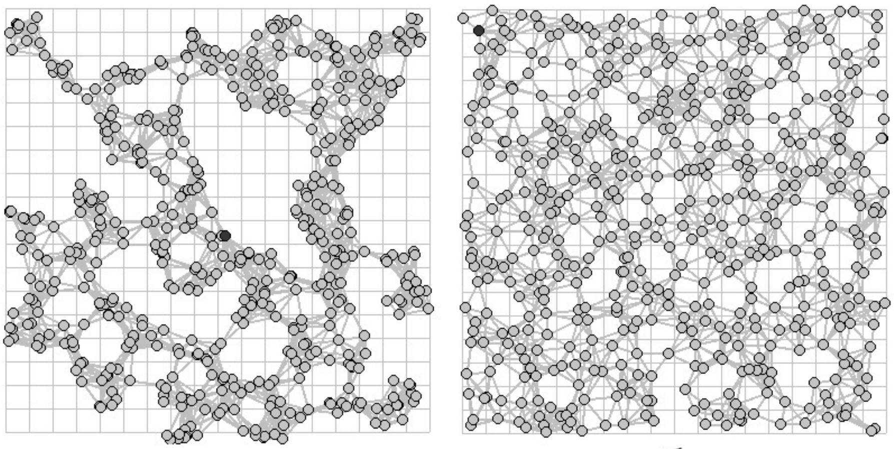
— вершины распределены равномерно в области действия сети;
— вся область покрывается кругами с центрами в вершинах графа и радиусами R;
— граф связен.

Кроме того, предлагаемый генератор превосходит аналоги по производительности.

2. Алгоритмы генерации случайных геометрических графов

Интуитивно понятный и традиционно используемый алгоритм генерации случайного геометрического графа [7] заключается в последовательном случайном размещении точек на плоскости, проверке возможности связности для каждой пары вершин и проверке заданных свойств полученной топологии. Количество вершин графа, радиус передачи, параметры области размещения вершин являются входными параметрами алгоритма. Для получения координат вершин применяют генератор псевдослучайных чисел с непрерывным равномерным распределением. Если полученный граф не обладает заданными свойствами, например не является связным, то он отбраковывается. Данная операция выполняется до тех пор, пока не будет получен граф с заданными свойствами. Очевидно, что количество итераций может быть весьма большим, особенно для получения связного графа с небольшим числом вершин, малым радиусом передачи, но большой областью размещения. Такой подход будем считать базовым (традиционным) и будем использовать результаты его работы для сравнения с работой других алгоритмов.

Другой метод генерации топологии для беспроводных сетей описан в [4]: алгоритм C-CRUG (Constrained Connected Random Unit Graphs). Основная идея состоит в том, что следующая вершина присоединяется к вершине с минимальной степенью. Алгоритм строит заранее связный граф, что существенно уменьшает количество итераций. Недостаток данного алгоритма в том, что построенные вершины «сгущаются» около некоторых вершин (рис. 2, *а*), что затрудняет адекватную проверку алгоритмов маршрутизации. Это ещё более заметно, когда начальная вершина располагается около границы области построения или в углу области. На каждом шаге для новой вершины требуется также проверка расстояния от неё до всех построенных вершин, что при большом числе вершин снижает эффективность алгоритма. Кроме того, алгоритм не гарантирует полного покрытия кругами (с центрами в вершинах графа и радиусами R) всей области действия сети, что существенно снижает его ценность.



а *б*

Рис. 2. Графы, сгенерированные алгоритмом C-CRUG (*а*) и традиционным алгоритмом (*б*)

3. Эффективный алгоритм генерации псевдослучайных геометрических графов

Предлагаемый далее эффективный метод генерации графов является развитием результатов, опубликованных в [13]. Быстродействие алгоритма достигается за счёт того, что изначально генерируются графы с требуемыми свойствами — связность и покрытие всей заданной области. Гарантированно не требуется проверка сгенерированного графа на корректность (иногда проверить те или иные свойства графа — достаточно трудозатратный процесс) и не выполняется процедура отбраковки графов, что позволяет значительно уменьшить количество вычислительных операций.

Повышение производительности генератора достигается за счёт построения вспомогательной квадратной сетки. Алгоритм легко адаптируется для случая, когда область размещения вершин не является квадратной.

На исследуемую область накладываем квадратную сетку (т. е. разбиваем квадрат на ячейки). Шаг сетки выбирается в зависимости от количества вершин графа (чем больше вершин, тем меньше шаг). Без ограничения общности можно считать, что область разделена на целое количество ячеек L_{size} по горизонтали и вертикали. Каждую ячейку будем обозначать $L[i,j]$, где i и j — места ячейки в сетке по горизонтали и вертикали. Рассмотрим сетки с такими вариантами для шагов: $L_{\text{step}} = \{R; R/\sqrt{2}; R/\sqrt{5}\}$.

Если шаг сетки $L_{\text{step}} = R$, то область действия сигнала из вершины (круг радиуса R) не покрывает всю ячейку и для проверки связности необходимо проверять расстояния до вершин, уже находящихся внутри ячейки. Данный случай не представляет особого интереса, но введение сетки даже с таким шагом позволит ускорить генерацию графа.

Для того чтобы вершина, находящаяся в узле ячейки, была достижима для наиболее удалённой от неё вершины в этой же ячейке (т. е. расстояние между ними было не более R), шаг сетки должен быть $R/\sqrt{2}$ (рис. 3,*а*). Аналогично, для того чтобы вершина, находящаяся в узле ячейки, была достижима для наиболее удалённой от неё вершины в соседней ячейке, шаг сетки должен быть $R/\sqrt{5}$ (рис. 3,*б*).

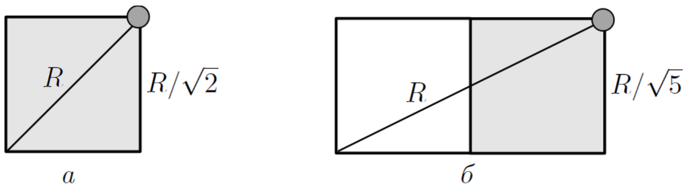


Рис. 3. Шаг сетки $L_{\text{step}} = R/\sqrt{2}$ (*а*) и $L_{\text{step}} = R/\sqrt{5}$ (*б*)

Чтобы обеспечить передачу данных из любой точки области развертывания БСС, необходимо выполнения условия: каждая ячейка сетки должна содержать хотя бы одну вершину. Исходя из этого, основная идея алгоритма состоит в следующем. Случайным образом (здесь и далее используем генератор равномерно распределённой случайной величины) бросаем по одной вершине в каждую ячейку. Затем выбираем случайным образом ячейку сетки и, случайно бросив в эту ячейку ещё одну вершину, проверяем расстояние от неё до вершин в смежных (псевдососедних) ячейках (рис. 4):

— для $L_{\text{step}} = R$ псевдососедними являются ячейки множества

$$\Omega_{i,j} = \{L[i+m, j+k] : m, k \in \{0, \pm 1\}\};$$

— для $L_{\text{step}} = R/\sqrt{2}$ псевдососедними являются ячейки множества

$$\Omega_{i,j} = \{L[i+m, j+k] : m, k \in \{0, \pm 1, \pm 2\}, 0 < |m| + |k| \leqslant 3\},$$

причём со всеми вершинами, уже находящимися в той же ячейке $L[i,j]$, новая вершина соединяется ребром без проверки расстояний;

— для $L_{\text{step}} = R/\sqrt{5}$ псевдососедними являются ячейки множества

$$\Omega_{i,j} = \{L[i+m, j+k] : m, k \in \{0, \pm 1, \pm 2, \pm 3\}, 1 < |m| + |k| \leqslant 4\},$$

причём со всеми вершинами в ячейках $|m| + |k| \leqslant 1$ новая вершина соединяется ребром без проверки расстояний (на рис. 4 такие ячейки отмечены тёмно-серым цветом).

Последнюю сгенерированную на каждом шаге вершину обозначим v_{new}. Проверка расстояний $\operatorname{dist}(v_{\text{new}}, v)$ выполняется только для вершин в выбранной ячейке и для вершин из смежных ячеек, что уменьшает количество операций, особенно при большом числе вершин n. Очевидно, что построенный граф является связным (кроме случая $L_{\text{step}} = R$, где необходимо делать дополнительные проверки на связность и, возможно, повторно генерировать вершины).

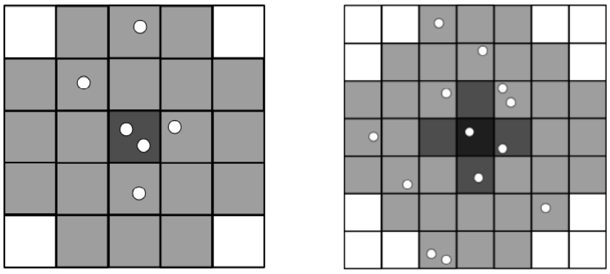


Рис. 4. Серым цветом отмечены псевдососедние ячейки для двух разных сеток ($L_{\text{step}} = R/\sqrt{2}$ и $R/\sqrt{5}$)

Заметим, что количество случайно размещённых вершин в ячейке можно описать биномиальным распределением с параметрами $N = n - L_{\text{size}}^2$, $p = L_{\text{size}}^{-2}$, где n — количество вершин в графе. Это свойство можно использовать для оптимизации алгоритма.

Пусть $B(N,p)$ — случайное число, полученное с помощью генератора псевдослучайных чисел для биноминально распределенной случайной величины с параметрами N и p. Будем последовательно бросать в каждую ячейку s вершин, где $s = B(N,p)$. Такой приём позволяет снизить число вычислительных операций, так как не требуется вычислять для каждой отдельной вершины соответствующую случайную ячейку. Краткая схема работы алгоритма приведена ниже (алгоритм 1). Поскольку количество вершин n обычно достаточно большое, можно вместо генератора биномиального распределения использовать генераторы пуассоновского или нормального распределения.

Алгоритм 1. Сетка

1: **Для** $i, j = 1, \ldots, L_{\text{size}}$
2: сгенерировать одну случайную вершину $v_{x,y}$ в ячейке $L[i,j]$;
3: проверить расстояние до вершин из множества ячеек $\Omega_{i,j}$ и соединить рёбрами те пары вершин, расстояния между которыми меньше либо равно R;
4: $n_{\text{curr}} := n - L_{\text{size}}^2$ — число несгенерированных вершин.
5: **Для** $i, j = 1, \ldots, L_{\text{size}}$
6: $s := B(N,p)$, $s_{i,j} := \min(s, n_{\text{curr}})$; сгенерировать $s_{i,j}$ вершин в ячейке $L_{i,j}$;
7: проверить расстояние до вершин из множества ячеек $\Omega_{i,j}$ и соединить рёбрами соответствующие вершины;
8: $n_{\text{curr}} := n_{\text{curr}} - s_{i,j}$.
9: **Если** $n_{\text{curr}} = 0$, **то**
конец алгоритма.

Поскольку в первой части алгоритма в каждой ячейке генерируется вершина, полученный граф связен и покрывает всю заданную область. Во второй части алгоритма генерация вершин даёт их равномерное распределение в заданной области. Следовательно, сгенерированный граф удовлетворяет требуемым условиям.

Если число вершин недостаточно для покрытия всей области, то описанный алгоритм можно адаптировать. Выбираем случайным образом пустую ячейку сетки $L[i,j]$, для которой в соседней с ней ячейке есть вершины, затем в выбранной ячейке случайным образом генерируем s вершин, где $s = B(n,p)$. Процесс заканчивается, когда все n вершин (сенсоров сети) распределены по ячейкам. В результате будет сформирован связный граф, удовлетворяющий заданным условиям, за исключением условия покрытия всей области. Краткая схема работы алгоритма приведена ниже (алгоритм 2). Для сравнения можно заметить, что в этом случае традиционный алгоритм [7] делает много операций отбраковки графов из-за их несвязности, что приводит к увеличению времени его работы.

Алгоритм 2. Сетка для небольшого числа вершин

1: Случайно выбрать ячейку, сгенерировать в ней $s_{i,j} = B(n,p) > 0$ вершин и соединить рёбрами те пары вершин, расстояние между которыми меньше либо равно R;
2: $n_{\text{curr}} := n - s_{i,j}$ — число несгенерированных вершин.
3: **Повторять**
4: случайно выбрать пустую ячейку $L[i,j]$, такую, что в соседней с ней ячейке есть вершины;
5: $s := B(n,p) > 0$, $s_{i,j} := \min(s, n_{\text{curr}})$; сгенерировать $s_{i,j}$ вершин в ячейке $L_{i,j}$;
6: проверить расстояния до вершин из множества ячеек $\Omega_{i,j}$ и соединить рёбрами соответствующие вершины (если граф получился несвязным, то повторно сгенерировать вершины в выбранной ячейке);
7: $n_{\text{curr}} := n_{\text{curr}} - s_{i,j}$.
8: **Пока** $n_{\text{curr}} > 0$.

4. Численные эксперименты

Программно реализованы следующие алгоритмы генерации случайных геометрических графов: традиционный алгоритм [7] и разработанный авторами алгоритм Сетка. Расчёты производились для различных исходных данных: количество вершин $|V|$ от 500 до 6000, шаг ячейки L_{step} равен R, $R/\sqrt{2}$, $R/\sqrt{5}$. Радиус $R = 30$, область размещения вершин 420×420. На рис. 5 представлено время (в секундах) генерации 100 случайных графов традиционным методом и вариациями предложенного алгоритма.

Численные расчёты показывают, что метод Сетка позволяет в среднем в 5–10 раз улучшить время генерации по сравнению с традиционным алгоритмом. При этом сгенерированные графы обладают схожими статистическими свойствами, что видно из гистограммы (рис. 6). Небольшое отклонение в графиках обусловлено тем, что традиционный алгоритм не гарантирует полное покрытие области.

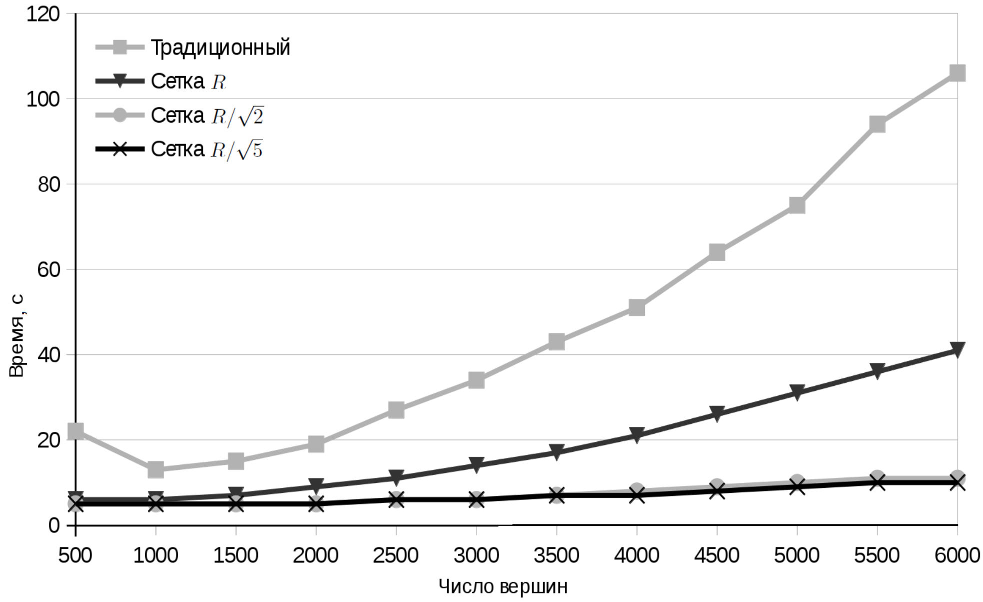


Рис. 5. Сравнение времени генерации 100 графов традиционным и предложенным методом Сетка с использованием биноминального распределения для сеток с шагом R, $R/\sqrt{2}$, $R/\sqrt{5}$

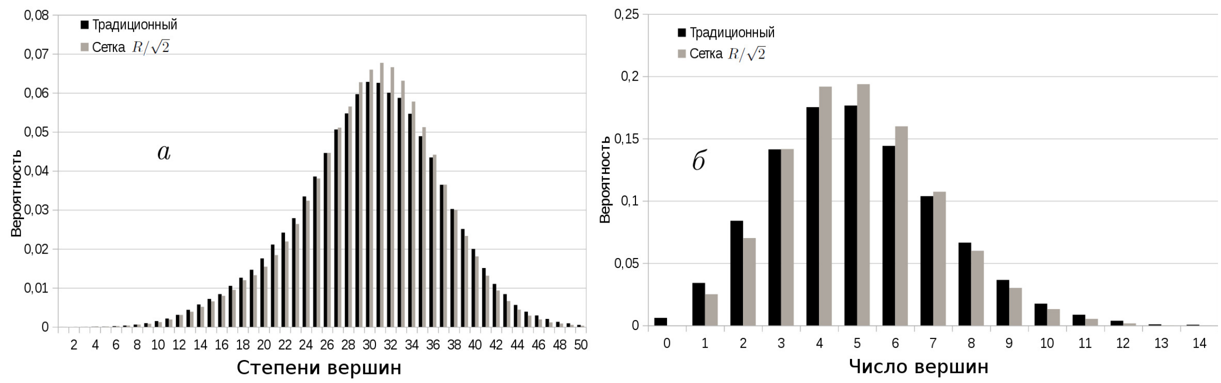


Рис. 6. Гистограммы степеней вершин (a) и числа вершин ($б$) в каждой ячейке для графов, сгенерированных традиционным методом и методом Сетка с использованием биноминального распределения ($|V| = 2000$, $L_{\text{step}} = R/\sqrt{2}$)

Заключение

Представленный генератор позволяет получать большое количество графов, моделирующих топологии беспроводных сетей, для анализа проектных решений. Разработанные методы генерации удобны также и для построения псевдослучайных геометрических графов с дополнительными ограничениями, такими, как ограничение на степень вершин, диаметр графа. Предложенные методы значительно лучше существующих, так как позволяют генерировать графы с наперёд заданными свойствами за меньшее время.

ЛИТЕРАТУРА

1. Algorithms for Sensor and Ad Hoc Networks / eds. D. Wagner and R. Wattenhofer. LNCS. 2007. V. 4621. 416 p.

2. *Penrose M. D.* Random Geometric Graphs. Oxford Studies in Probability. Oxford University Press, 2003.
3. *Clark B. N., Colbourn C. J., and Johnson D. S.* Unit disk graphs // Discr. Math. 1990. V. 86(13). P. 165–177.
4. *Onat F. A., Stojmenovic I., and Yanikomeroglu H.* Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks // Pervasive and Mobile Computing. 2008. V. 4. Iss. 5. P. 597–615.
5. *Guo Y., Duan L., and Zhang R.* Optimal pricing and load sharing for energy saving with cooperative communications // IEEE Trans. Wireless Communications. 2016. V. 15. Iss. 2. P. 951–964.
6. *Paris S., Nita-Rotaru C., Martignon F., and Capone A.* Cross-layer metrics for reliable routing in wireless mesh networks // IEEE/ACM Trans. Networking. 2013. V. 21. Iss. 3. P. 1003–1016.
7. *Seo J., Kim M., Hur I., et al.* DRDT: Distributed and Reliable Data Transmission with cooperative nodes for lossy wireless sensor networks // Sensors. 2010. V. 10(4). P. 2793–2811.
8. *Цициашвили Г. Ш., Осипова М. А., Лосев А. С.* Асимптотика вероятности связности графа с низконадёжными рёбрами // Прикладная дискретная математика. 2013. № 1. С. 93–98.
9. *Мелентьев В. А.* Аналитический подход к синтезу регулярных графов с заданными значениями порядка, степени и обхвата // Прикладная дискретная математика. 2010. № 2. С. 74–86.
10. *Camilo T., Silva J. S., Rodrigues A., and Boavida F.* GENSEN: A topology generator for real wireless sensor networks deployment // LNCS. 2007. V. 4761. P. 436–445.
11. *Гавриленко В. Г., Ельцов А. Ю., Конюченко А. В. и др.* Разработка симулятора сенсорных сетей с детальным моделированием физического уровня в среде ANYLOGIC // Имитационное моделирование. Теория и практика: Сб. докл. Второй Всерос. науч.-практич. конф. ИММОД-2005. Т. 1. СПб.: ЦНИИТС, 2005. С. 182–185.
12. *Zhou C. and Krishnamachari B.* Localized topology generation mechanisms for wireless sensor networks // IEEE GLOBECOM' 03, San Francisco, CA, December 2003. P. 1269–1273.
13. *Shakhov V. V., Sokolova O. D., and Yurgenson A. N.* A fast method for network topology generating // LNCS. 2014. V. 8715. P. 96–101.

REFERENCES

1. Algorithms for Sensor and Ad Hoc Networks, eds. D. Wagner and R. Wattenhofer, LNCS, 2007, vol. 4621, 416 p.
2. *Penrose M. D.* Random Geometric Graphs. Oxford Studies in Probability. Oxford University Press, 2003.
3. *Clark B. N., Colbourn C. J., and Johnson D. S.* Unit disk graphs. Discr. Math., 1990, vol. 86(13), pp. 165–177.
4. *Onat F. A., Stojmenovic I., and Yanikomeroglu H.* Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks. Pervasive and Mobile Computing, 2008, vol. 4, iss. 5, pp. 597–615.
5. *Guo Y., Duan L., and Zhang R.* Optimal pricing and load sharing for energy saving with cooperative communications. IEEE Trans. Wireless Communications, 2016, vol. 15, iss. 2, pp. 951–964.
6. *Paris S., Nita-Rotaru C., Martignon F., and Capone A.* Cross-layer metrics for reliable routing in wireless mesh networks. IEEE/ACM Trans. Networking, 2013, vol. 21, iss. 3, pp. 1003–1016.

7. *Seo J., Kim M., Hur I., et al.* DRDT: Distributed and Reliable Data Transmission with cooperative nodes for lossy wireless sensor networks. Sensors, 2010, vol. 10(4), pp. 2793–2811.
8. *Tsitsiashvili G. Sh., Osipova M. A., and Losev A. S.* Asimptotika veroyatnosti svyaznosti grafa s nizkonadezhnymi rebrami [Asymptotics for connectivity probability of graph with low reliable arcs]. Prikladnaya Diskretnaya Matematika, 2013, no. 1, pp. 93–98. (in Russian)
9. *Melent'ev V. A.* Analiticheskiy podkhod k sintezu regulyarnykh grafov s zadannymi znacheniyami poryadka, stepeni i obkhvata [An analytical approach to the synthesis of regular graphs with preset values of the order, degree and girth]. Prikladnaya Diskretnaya Matematika, 2010, no. 2, pp. 74–86. (in Russian)
10. *Camilo T., Silva J. S., Rodrigues A., and Boavida F.* GENSEN: A topology generator for real wireless sensor networks deployment. LNCS, 2007, vol. 4761, pp. 436–445.
11. *Gavrilenko V. G., El'tsov A. Yu., Konyuchenko A. V. et al.* Razrabotka simulyatora sensornykh setey s detal'nym modelirovaniem fizicheskogo urovnya v srede ANYLOGIC [The development of sensor network simulator with a detailed simulation of the physical level in the ANYLOGIC environment]. Imitatsionnoe Modelirovanie. Teoriya i Praktika, Proc. IMMOD-2005, vol. 1. St. Petersburg, SSTC Publ., 2005, pp. 182–185. (in Russian)
12. *Zhou C. and Krishnamachari B.* Localized topology generation mechanisms for wireless sensor networks. IEEE GLOBECOM' 03, San Francisco, CA, December 2003, pp. 1269–1273.
13. *Shakhov V. V., Sokolova O. D., and Yurgenson A. N.* A fast method for network topology generating. LNCS, 2014, vol. 8715, pp. 96–101.