5. Хорошун Л. П. Статистическая механика и эффективные свойства материалов / Л. П. Хорошун, Б. П. Маслов, Е. Н. Шикула, Л. В. Назаренко // Механика композитов : В 12 томах, Т. 3. – К. : Наукова думка, 1993. – 388 с.

6. Гребенюк С. М. Визначення ефективного модуля пружності композиту при нормальному розподілі модулів пружності волокна та матриці / С. М. Гребенюк, М. І. Клименко // Вестник Херсонского национального университета. – 2014. – № 3(50). – С. 254-258.

7. Hrebeniuk S. Effective elastic modulus determination of unidirectional composite for stochastic geometric characteristics of fiber / S. Hrebeniuk, M. Klymenko, K. Omelchenko // Вісник Запорізького національного університету. Фізико-математичні науки. – 2014. – № 1. – С. 14-23.

REFERENCES

1. Rasteryaev, JU.K. and Agal'cov, G.N. (2005), "Rubber-cord Composite materials and their mechanical deformation", *Geotehnichna mehanika*, issue 60, pp. 200-248.

2. Klastorny, M., Konderla, P. and Piekarskiy, R. (2009), "Exact theory stiffness of unidirectional fiber-reinforced composites", *Mehanika kompozitnyh materialov*, vol. 45, no. 1, pp. 109-144.

3. Grebenjuk, S.N. (2011), "Elastic characteristics of the composite material with the matrix and fiber transtropic", *Metody rozv"yazuvannya prykladnykh zadach mekhaniky deformivnoho tverdoho tila*, issue 12, pp. 62-68.

4. Grebenjuk, S.N. (2012), "Definition of the shear modulus of the composite material with the matrix and fiber transtropic", *Metody rozv"yazuvannya prykladnykh zadach mekhaniky deformivnoho tverdoho tila*, issue 13, pp. 92-98.

5. Horoshun, L.P., Maslov, B.P., SHikula, E.N. and Nazarenko, L.V. (1993), *Statisticheskaya mekhanika i effektivnye svoystva materialov* [Statistical mechanics and the effective properties of materials], Naukova dumka, Kiev.

6. Grebenjuk, S.M. and Klimenko, M.I. (2014), "Determination of the effective modulus composite normal distribution elastic modulus fibers and the matrix", *Vestnik Hersonskogo nacional'nogo universiteta*, no. 3(50), pp. 254-258.

7. Hrebeniuk, S., Klymenko, M. and Omelchenko, K. (2014), "Effective elastic modulus determination of unidirectional composite for stochastic geometric characteristics of fiber", *Visnik Zaporizkogo natsionalnogo universitetu*, *Fizyko-matematychni nauky*, no. 1, pp. 14-23.

УДК 519.8

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ГРАФОВ С ЗАДАННЫМИ СВОЙСТВАМИ

Козин И. В., д. ф.-м. н., профессор, Батовский С. Е., аспирант, Сардак В. И., аспирант

Запорожский национальный университет,
ул. Жуковского, 66, г. Запорожье, 69600, Украина

ainc00@gmail.com, user.sergey.b@gmail.com

Исследуется проблема генерации графов для создания баз данных тестовых задач. Рассматриваются алгоритмы генерации случайных графов с различными свойствами. Для различных видов графов приводятся алгоритмы генерации (граф-генераторы). Предложены граф-генераторы для деревьев произвольного вида, для деревьев с ограничениями на степени вершин, для графов произвольного вида, для связных графов, для графов с ограничениями на степени вершин, для регулярных графов, для графов с заданным числом ребер.

Ключевые слова: случайные графы, алгоритмы генерации графов, модель Эрдеша-Реньи, связный граф, регулярный граф.

ГЕНЕРАЦІЯ ВИПАДКОВИХ ГРАФІВ ІЗ ЗАДАНИМИ ВЛАСТИВОСТЯМИ

Козін І. В., к. ф.-м. н., професор, Батовський С. Є., аспірант, Сардак В. І., аспірант

Запорізький національний університет,
вул. Жуковського, 66, м. Запоріжжя, 69600, Україна

ainc00@gmail.com, user.sergey.b@gmail.com

Досліджується проблема генерації графів для створення бази даних тестових задач. Розглядаються алгоритми генерації випадкових графів з різними властивостями. Для різних видів графів наводяться алгоритми генерації – (граф-генератори). Запропоновано граф-генератори для довільних дерев, для дерев з обмеженням на степені вершин, для довільних графів, для зв'язних графів, для графів з обмеженням на степені вершин, для регулярних графів, для графів із заданим числом ребер.

Ключові слова: випадкові графи, алгоритми генерації графів, алгоритм Ердеша-Реньї, зв'язний граф, регулярний граф.

GENERATION OF RANDOM GRAPHS WITH DESIRED PROPERTIES

Kozin I. V., Dr. Sc. (Phys.-Math.), Batovskyi S. Ye., postgraduate, Sardak V. I., postgraduate

Zaporizhzhya State University,
Zhukovsky str., 66, Zaporizhzhya, 69600, Ukaine

ainc00@gmail.com, user.sergey.b@gmail.com

The article considers generalizations of Erdos-Renyi model for the generation of random graphs with certain properties. In particular, it provides a method for generating trees, connected graphs, graphs with restrictions on the degree of vertices, regular graphs, graphs with a given number of edges.

These methods are used in the programs, also known as graph-generators, that simulate the known NP-complete problems on graphs. Examples of such problems may be: traveling salesman problem, the problem of the maximum clique, problem of covering a graph by stars, etc. That is why every algorithm in the article is accompanied by an example of its implementation by pseudocode.

Each random graphs generation algorithm, which described in this article, can be easily modified for generation weighted line graphs with random weights of the specified range. Also, the authors recommend the use these algorithms for creating a large database of test problems, which can be used in further for the comparative statistical analysis of various algorithms of exact or approximate optimization on graphs and networks. Because the such analysis is practically the only way to check the quality of different types of metaheuristics.

Keywords: random graphs, algorithms of graph generating, Erdos-Renyi algorithm, connected graph, regular graph.

ВВЕДЕНИЕ. ПОСТАНОВКА ПРОБЛЕМЫ

Большое количество прикладных оптимизационных задач на сегодняшний день не могут быть точно решены, поскольку их вычислительная сложность относится к классу NP-трудных [1]. Однако потребность в решении таких задач возрастает и, соответственно, увеличивается поток работ, посвященных поискам приближенных решений трудных задач. Во многих случаях для поиска приближенных решений используются метаэвристики различных типов. Однако при использовании метаэвристик открытым остается вопрос о качестве предлагаемого метода. Поскольку эвристические алгоритмы не имеют априорных оценок сходимости, то неизвестно насколько точно полученное приближённое решение соответствует оптимальному. Как следствие, очень трудно оценить целесообразность применения алгоритма в различных оптимизационных задачах.

Одним из возможных решений данной проблемы может служить проверка метаэвристических алгоритмов на примерах из известных тестовых библиотек с известными рекордами, и попытка улучшить эти рекорды. Одной из таких библиотек является библиотека ORLIB [2].

Другим подходом к решению проблемы оценки качества алгоритмов является сравнение «нового» алгоритма с другими алгоритмами, работа которых уже детально изучена. Такое

сравнение должно проходить на большой базе тестовых задач со случайным набором начальных данных. Это позволит вычислить «средние» характеристики эффективности тестируемого алгоритма.

Многие интересные труднорешаемые задачи могут быть сведены к известным задачам на графах, к таким, как задача коммивояжёра, задача о максимальной клике, задача о покрытии графа звёздами и т.д. В каждой такой задаче условием является некоторый граф (возможно взвешенный). Разумно проводить сравнение алгоритмов на классах графов с заданными свойствами. Таким образом, для создания набора тестовых задач возникает необходимость разработки специальных алгоритмов генерации случайных графов с заданными свойствами и создании соответствующих компьютерных программ – граф-генераторов.

ОБЗОР ИМЕЮЩИХСЯ РЕЗУЛЬТАТОВ

Случайные графы являются объектом пристального внимания исследователей с 70-х годов прошлого века [3-6]. В настоящее время в связи с широким применением теории графов при описании и моделировании объектов различного типа, сетевых структур актуальным является вопрос о создании имитационных моделей на основе генерации графов с заданными свойствами. Кроме того, генераторы случайных графов необходимы при проверке статистических свойств эвристических алгоритмов на графах. Наиболее известной моделью генератора случайных графов является модель Эрдёша-Реньи [7], в которой каждое ребро графа генерируется с одинаковой заданной вероятностью. Более поздние исследования в области генерации графов связаны с моделированием сетевых структур с заданными свойствами, в частности, с моделированием www-сетей. К таким моделям относятся модель Барабаши-Альберта [8], модель Боллобаша-Риордана [9], а также ряд других [10].

ФОРМУЛИРОВКА ЦЕЛЕЙ

Целью настоящей работы является исследование генераторов случайных графов с заданными топологическими свойствами (связность, степени вершин, наличие цикла и т.д.) на основе обобщения классической модели Эрдеша-Реньи и описание соответствующих алгоритмов.

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ДЕРЕВЬЕВ

Рассмотрим задачу генерации случайного дерева с заданным числом вершин. Напомним, что деревом называется связный граф без циклов. Таким образом, в n-вершинном дереве имеется ровно $n-1$ ребер. Граф будем описывать матрицей смежности $C=\left(c_{ij}\right)_{i,j=1,2,...,n}$, в которой

$$c_{ij}=\begin{cases}1, & \text{если вершины } i \text{ и } j \text{ соединены ребром,}\\ 0, & \text{если вершины } i \text{ и } j \text{ не соединены ребром.}\end{cases}$$

Вершины графа последовательно индексируются числами от 1 до n. Все полученные индексы разделяются на два множества I_1 и I_2 так, что на начальном этапе $I_1=\{\ \}$, $I_2=\{1,2,...,n\}$. На первом шаге выбирается случайная пара индексов и переносятся из множества I_2 в I_1. Это соответствует выбору первого ребра дерева. На очередном шаге процесса из каждого множества случайно выбирается по одному индексу: $i_1\in I_1$ и $i_2\in I_2$. В граф добавляется ребро, которое соединяет вершины с выбранными индексами. После чего индекс i_2 удаляется из множества I_2 и заносится в множество I_1. Процесс продолжается пока множество I_2 не окажется пустым. Простейшая реализация алгоритма представлена на рис. 1.

```
var I1[n] = { };
var I2[n] = {1, 2, 3, …, n };

   // случайно выбирается ребро
   var i1 = I2[random(1, I2.Length)];
   I2.Remove(i1);
   var i2 = I2[random(1, I2.Length)];
   I1.Add(i1);
   I1.Add(i2);
   c[i1, i2] = 1;
do
{
   var i1 = I1[random(1, I1.Length)];
   var i2 = I2[random(1, I2.Length)];

   // ребро добавляется в граф
   c[i1, i2] = 1;

   I2.Remove(i2);
   I1.Add(i2);
}
while (I2.Length != 0);
```

Рис. 1. Алгоритм генерации случайного дерева

Результатом работы алгоритма является матрица смежности, соответствующая случайному n-вершинному дереву.

МОДЕЛЬ ЭРДЕША-РЕНЬИ

Одним из простейших алгоритмов для генерации случайных графов является алгоритм на основе модели Эрдёша-Реньи [10]. Алгоритм очень прост в использовании, поскольку имеет всего лишь один параметр $\alpha \in (0,1)$, который задаёт вероятность появления произвольного ребра в графе. Применяя данный алгоритм, вероятность получить граф $G = (V, E)$ одинакова для всех графов с заранее заданным числом вершин n и числом ребер m:

$$P(G) = \alpha^m (1-\alpha)^{\frac{n(n-1)}{2}-m}.$$

Алгоритм Эрдёша-Реньи формирует граф, заполняя его матрицу смежности $C = (c_{ij})_{i,j=1,2,\ldots,n}$. Ниже приведён вариант реализации данного алгоритма (рис. 2).

```
for (var i = 1; i <= n; i++)
{
   for (var j = i + 1; j <= n; j++)
   {
      // random(x, y) возвращает случайное число на отрезке [x, y]
      if (random(0, 1) <= alpha)
      {
         c[i, j] = 1;
      }
      else
      {
         c[i, j] = 0;
      }
   }
}
```

Рис. 2. Алгоритм Эрдёша-Реньи

Случайный граф, сгенерированный на основе модели Эрдёша-Реньи, не обязательно является связным. Одним из способов генерации, гарантирующих связность результата генерации, может быть следующий: сначала генерируется остовное дерево с заданным числом вершин, а

затем к этому дереву по алгоритму Эрдёша-Реньи добавляются ребра, не принадлежащие дереву. То есть каждое новое ребро добавляется с одной и той же вероятностью α.

Рассмотрим теперь задачу генерации случайного графа, содержащего гамильтонов цикл. Гамильтонов цикл однозначно определяется некоторой перестановкой вершин графа. Для генерации случайной перестановки можно воспользоваться тасованием Фишера-Йетса [11]. Одна из возможных реализаций данного алгоритма приведена ниже (рис. 3).

```
var a[n] = {1, 2, …, n};
for (var i = 1; i <= n; i++)
{
   // round(x) округляет число x до ближайшего целого
   var j = 1 + round(random(0, 1) * (n - 1));

   // swap(x, y) обменивает значениями переменные x и y
   swap(a[i], a[j]);
}
```

Рис. 3. Тасование Фишера-Йетса

На основе полученной случайной перестановки выполняется генерация гамильтонова цикла в графе путём заполнения его матрицы смежности $C = \left(c_{ij}\right)_{i,j=1,2,\ldots,n}$ (рис. 4).

```
c[a[n], a[1]] = 1;
for (var i = 1; i <= n - 1; i++)
{
   c[a[i], a[i + 1]] = 1;
}
```

Рис. 4. Процесс заполнения матрицы смежности для построения гамильтонова цикла

Затем, чтобы построить случайный граф, необходимо добавить по алгоритму Эрдёша-Реньи ребра, не принадлежащие циклу.

ГРАФЫ С ОГРАНИЧЕНИЕМ НА СТЕПЕНИ ВЕРШИН

Рассмотрим теперь задачу генерации случайных графов, у которых степень каждой вершины не превосходит заданного целого числа $d \le n-1$. Такой граф может быть построен с помощью модифицированного алгоритма Эрдёша-Реньи (рис. 5), который использует дополнительный массив D степеней вершин.

```
// вначале все элементы массива степеней вершин равны нулю
var D[n] = {0, 0, …, 0};
for (var i = 1; i <= n; i++)
{
   for (var j = i + 1; j <= n; j++)
   {
      if ((D[i] <= d) and (D[j] <= d))
      {
         if (random(0, 1) <= alpha)
         {
            c[i, j] = 1;
            D[i]++;
            D[j]++;
         }
         else
         {
            c[i, j] = 0;
         }
      }
   }
}
```

Рис. 5. Алгоритм Эрдёша-Реньи, учитывающий ограничение на степень вершин сверху

При необходимости генерации связного графа, также, модифицируется и процесс построения случайного остовного дерева (рис. 6).

```
…
   I2.Remove(i2);
   I1.Add(i2);

   if (D[i1] == d) // добавлена проверка для D[i1]
   {
      I1.Remove(i1);
   }

   if (D[i2] == d) // добавлена проверка для D[i2]
   {
      I1.Remove(i2);
   }
…
```

Рис. 6. Изменённая часть алгоритма генерации случайного остовного дерева, учитывающая ограничение на степень вершин сверху

Вершины графа, степени которых уже достигли установленного максимума, вовсе не участвуют в процессе. Также, на каждом шаге алгоритма после добавления нового ребра (i_1, i_2) значения $D[i_1]$ и $D[i_2]$ увеличиваются на единицу, причём, если $D[i_k],\ k \in \{1,2\}$ достигает значения d, то соответствующий индекс i_k удаляется из своего множества.

Генерация произвольного случайного графа с ограничением на степень вершин снизу некоторым положительным числом $d \le n-1$ может быть выполнена приведённым выше усовершенствованным алгоритмом Эрдёша-Реньи (рис. 5). В этом случае, вначале строится случайный граф с ограничением на степень вершин сверху числом $n-1-d$. После чего берётся его дополнение до полного n-вершинного графа.

Алгоритм несколько усложняется при построении связного графа. Весь процесс генерации происходит в три этапа. Вначале строится случайное остовное дерево с матрицей смежности $C = (c_{ij})_{i,j=1,2,\ldots,n}$ и массивом степеней вершин D_1. Затем происходит формирование дополнения случайного графа с матрицей смежности $C_{dop} = \left(c_{ij}^{dop}\right)_{i,j=1,2,\ldots,n}$ по следующему изменённому алгоритму Эрдёша-Реньи (рис. 7).

```
var D[n] = {0, 0, …, 0};
for (var i = 1; i <= n; i++)
{
   for (var j = i + 1; j <= n; j++)
   {
      if ((D[i] <= (n – 1 – d + D1[i])) and (D[j] <= (n – 1 – d + D1[j])))
      {
         if (random(0, 1) <= alpha)
         {
            cdop[i, j] = 1;
            D[i]++;
            D[j]++;
         }
         else
         {
            cdop[i, j] = 0;
         }
      }
   }
}
```

Рис. 7. Специализированный алгоритм Эрдёша-Реньи

После этого формируется матрица смежности искомого случайного графа (рис. 8).

```
for (var i = 1; i <= n; i++)
{
   for (var j = i + 1; j <= n; j++)
   {
      // min(x, y) – возвращает меньшее из двух чисел x и y
      c[i, j] = min(1, 1 – cdop[i, j] + t[i, j]);
   }
}
```

Рис. 8. Процесс формирования матрицы смежности графа, с ограничением на степень вершин снизу

В результате будет получен случайный связный граф, степень каждой вершины которого будет не менее заданного числа d.

Рассмотрим теперь задачу генерации регулярного графа. Пусть степень каждой вершины графа равна d (в этом случае произведение nd обязательно является четным). Пусть также вероятность появления недостающего ребра $\alpha = 1$. Тогда в результате работы алгоритма, листинг которого представлен на рис. 5, всегда будет получаться случайный регулярный граф, степени вершин которого равны d.

СЛУЧАЙНЫЕ ГРАФЫ С ЗАДАННЫМ ЧИСЛОМ РЕБЕР

В заключение рассмотрим задачу генерации случайного графа с заданным числом вершин n и заданным числом ребер m. Каждому возможному ребру $(i, j)_{i=2,\ldots,n-1;\, j<i}$ сопоставим индекс $k = n*(i-1)+j$ этого ребра, однозначно соответствующий ребру. Из массива индексов случайным образом выберем m различных индексов и соответствующие им пары чисел (i, j) будем рассматривать как ребра случайного графа. Алгоритм легко модифицируется для генерации связного графа с заданным количеством вершин и ребер.

ВЫВОДЫ

В данной статье были рассмотрены обобщения модели Эрдеша-Реньи для генерации случайных графов с определенными свойствами. В частности, были приведены методы для генерации деревьев, связных графов, графов с ограничениями на степени вершин, регулярных графов, графов с заданным числом ребер.

Любой из описанных в статье алгоритмов генерации случайных графов может быть легко модифицирован для генерации реберно-взвешенных графов со случайными весами ребер из заданного диапазона весов. Алгоритмы генерации графов позволяют создавать большие базы тестовых задач, которые можно использовать для сравнительного статистического анализа различных точных и приближенных алгоритмов оптимизации на графах и сетях. Такой анализ является практически единственным методом проверки качества метаэвристик различных типов.

ЛИТЕРАТУРА

1. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон ; пер. А. Фридман. – М. : Мир, 1982. – 416 с.

2. Beasley J. E. OR-Library: distributing test problems by electronic mail / J. E. Beasley // Journal of the Operational Research Society. – 1990. – № 41. – P. 1069-1072.

3. Erdos P. Probabilistic methods in combinatorics / P. Erdos, J. Spencer. – New York : Academic Press, 1974. – P. 1375-1383.

4. Степанов В. Е. Комбинаторная алгебра и случайные графы / В. Е. Степанов // Теория вероятностей и ее применение. – 1969. – № 14:3. – С. 393-420.

5. Коваленко И. Н. К теории случайных графов / И. Н. Коваленко // Кибернетика. – 1971. – № 4. – С. 1-4.

6. Колчин В. Ф. Случайные графы / В. Ф. Колчин. – М. : ФИЗМАТЛИТ, 2004. – 2-е изд. – 256 с.

7. Erdös P. On the evolution of random graphs / P. Erdös, A. Rényi // Publ. Math. Debrecen. – 1959. – Vol. 6. – P. 290-297.

8. Barab´asi L.-A. Emergence of scaling in random networks / L.-A. Barab´asi, R. Albert // Science. – 1999. – № 286. – P. 509-512.

9. Райгородский А. М. Модели случайных графов и их применения / А. М. Райгородский // Труды МФТИ. – 2010. – Т. 2, № 4. – С. 130-140.

10. Берновский М. М. Случайные графы, модели и генераторы безмасштабных графов / М. М. Берновский, Н. Н. Кузюрин // Труды Института системного программирования РАН. – 2012. – Т. 22. – С. 419-432.

11. Кнут Д. Искусство программирования. Т. 2. Получисленные методы / Д. Кнут. – М. : «Вильямс», 2007. – 3-е изд. – 832 с.

REFERENCES

1. Gerry, M. and Jhonson, D. (1982), *Vychislitel'nye mashiny i trudnoreshaemye zadachi* [Computing machines and hard solvable problems], Translated by Fridman, A., Mir, Moskow, Russia.

2. Beasley, J.E. (1990), "OR-Library: distributing test problems by electronic mail", *Journal of the Operational Research Society*, no. 41, pp. 1069-1072.

3. Erdos, P. and Spencer, J. (1974), "Probabilistic methods in combinatorics", Academic Press, New York.

4. Stepanov, V.Ye. (1969), "Combinatorial algebra and random graphs", *Teoria veroyatnostey i yeyo primenenie,* no. 14:3, pp. 393-420.

5. Kovalenko, I.N. (1971), "To theory of random graphs", *Kibernetika*, no. 4, pp. 1-4.

6. Kolchin, V.F. (2004), *Sluchaynye grafy* [Random graphs], FIZMATLIT, Moskow, Russia.

7. Erdös, P. and Rényi, A. (1959). "On the evolution of random graphs", *Publ. Math. Debrecen*, vol. 6, pp. 290-297.

8. Barab´asi, L.-A. and Albert, R. (1999), "Emergence of scaling in random networks", *Science*, no. 286, pp. 509-512.

9. Raigorodskiy, A.M. (2010), "Models of random graphs and they usage", *Trudy MFTI*, vol. 2, no. 4, pp. 130-140.

10. Bernovskiy, M.M. and Kuzyurin, N.N. (2012), "Random graphs, models and generators of scale-free graphs", *Trudy instituta sistemnogo programmirovaniya RAN*, vol. 2, pp. 419-432.

11. Knuth, D. (2007), *Iskusstvo programmirovaniya. T. 2. Poluchislennye metody* [The Art of Computer Programming. Vol. 2. Seminumerical algorithms], Vil'yams, Moskow, Russia.