

## АЛГОРИТМЫ ПОСЛОЙНОЙ ГЕНЕРАЦИИ РЕГУЛЯРНЫХ ГРАФОВ

**Введение.** Генерация регулярных неизоморфных графов является проблемой конструктивной комбинаторики [1]. Помимо научного интереса, генерация таких графов имеет большое практическое значение. Регулярные графы применяются в разнообразных областях. Например, при идентификации молекул веществ, при разработке баз знаний интеллектуальных систем, в моделях механического вращения, в пиринговых сетях (некоторые из которых базируются на связанных регулярных неориентированных графах). Графы также применяются в информационных системах, развитие которых привело к их объединению в локальные и глобальные компьютерные сети. Это потребовало оптимизации топологии соединений. К примеру, когда необходимо наилучшим образом связать большое количество однородных элементов, например, концентраторов. Для такой оптимизации регулярные графы являются наилучшим инструментом.

**1. Постановка задачи.** Известны различные алгоритмы генерации регулярных графов [2,3]. Они делятся на точные (алгоритм прямого произведения множеств, алгоритм дерева [4] и др.) и приближенные (генетический алгоритм). Точные алгоритмы обеспечивают генерацию всех возможных графов с заданными параметрами, но отличаются большой трудоемкостью. Приближенные алгоритмы выполняются гораздо быстрее, но не гарантируют генерацию всех возможных графов. Однако одной из основных проблем генерации графов оказалось не само их создание, а отбор неизоморфных гра-

фов. Т.к. все алгоритмы генерируют графы с большой избыточностью. Существующие алгоритмы поиска изоморфизма (метод анализа вектора степеней вершин, метод поиска с возвратом и т.д.) в большинстве своем не пригодны для регулярных графов.

Помимо уже отмеченной классификации алгоритмов на точные и приближенные необходимо ввести еще один критерий деления: прямые и послойные алгоритмы. К прямым относятся алгоритмы дерева, прямого произведения множеств [4] и алгоритм циклического перебора (описанный ниже). Все эти алгоритмы отличаются высокой трудоемкостью, т.к. генерация очередного комплекта регулярных графов  $K_n^m$  (здесь  $n$  – число вершин графа,  $m$  – степень вершин) начинается с нуля.

В этом плане послойная генерация выгодно отличается тем, что очередная серия регулярных графов следующего порядка ( $K_{n+2}^m$ ) строится на базе уже ранее сгенерированного комплекта неизоморфных графов  $K_n^m$ . Такой способ построения еще можно назвать бесповторным.

Целью исследования является разработка алгоритмов быстрой генерации с небольшим числом изоморфных копий.

**2. Алгоритмы генерации регулярных графов циклическим перебором (прямой метод).** Для генерации регулярных графов можно

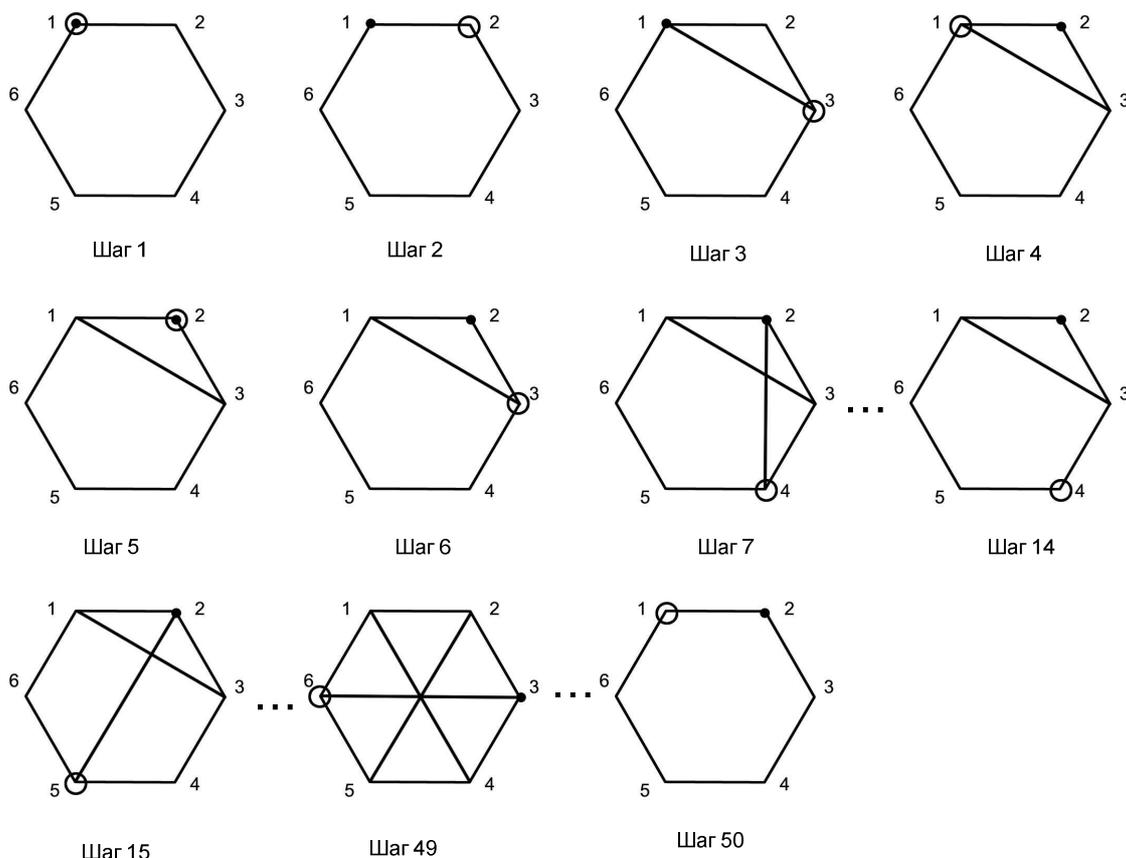


Рис. 1. Последовательность графов, генерируемых алгоритмом циклического перебора

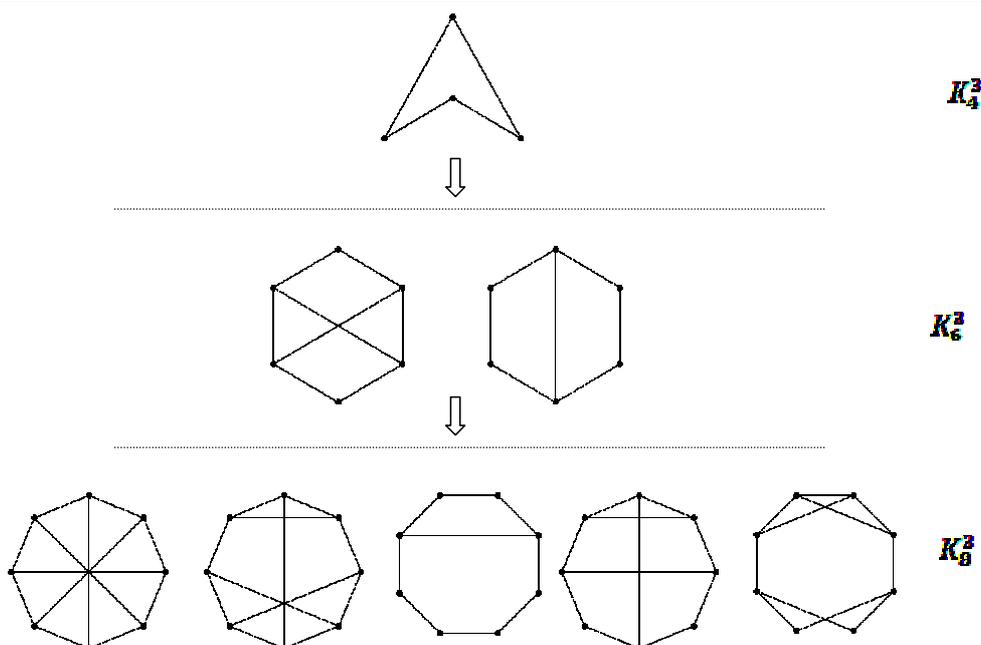


Рис. 2. Последовательная генерация графов  $K_8^3$

использовать наиболее очевидный алгоритм циклического перебора. На рисунке 1 пошагово представлен принцип его действия. Как и во всех прямых алгоритмах, первоначально строится обод-граф

$K_n^2$ . После расстановки в матрице смежности единиц таким образом, чтобы получить начальный обод, алгоритм начинает работу. Работа происходит с двумя вершинами, относительно которых вычисляется возможность, либо невозможность постановки на них ребра. На первом шаге эти вершины совпадают (рис. 1). Ребро поставить нельзя. Тогда вторая вершина смещается на единицу. Ребро 1-2 уже существует, поэтому на третьем шаге вершина смещается еще на единицу. Постановка ребра 1-3 допустима. После того, как ребро устанавливается, рекурсивно вызывается та же процедура, но с увеличенной на единицу первой вершиной и уменьшенной на единицу количеством свободных ребер. При этом вторая вершина возвращается на начало и опять последовательно проходит все вершины пока не найдет место, где возможна установка ребра.

На шаге 7 видно, что сложившаяся расстановка неразрешима, т.к. остается еще одно свободное ребро, которое поставить некуда. В этом случае алгоритм выходит на один шаг из рекурсии, что ведет к отмене последнего проставленного ребра (см. шаг 14) и нахождению другого места для постановки ребра. На шаге 49 был построен один из графов. Этот граф запоминается, первая точка переходит на следующую позицию, вторая ставится на начальную вершину и алгоритм повторяется. Пока первая точка не пройдет таким образом все вершины контура.

**3. Алгоритм установки парных вершин на ребре.** Основан на послойной генерации множества графов  $K_n^m$  из уже сгенерированного множества графов  $K_{n-2}^m$ . Общий принцип последовательной генерации на примере множества графов  $K_8^3$  изображен

на рисунке 2. Из рисунка видно, что на каждом новом слое количество вершин исходного графа увеличивается на две, степень вершин не изменяется. Поскольку с увеличением количества вершин, растет число комбинаций расстановки внутренних ребер, то количество сгенерированных графов тоже увеличивается.

Метод установки парных вершин на ребре представляет собой следующую последовательность действий:

На одном из внешних ребер устанавливаются две вершины, т.е. в обод добавляется одно новое ребро. Затем выбирается внутрен-

нее ребро и удаляется из графа. А из вершин, которые оно соединяло, проводятся ребра к новым вершинам, созданным на предыдущем шаге.

Результат проделанных действий изображен на рисунке 3.

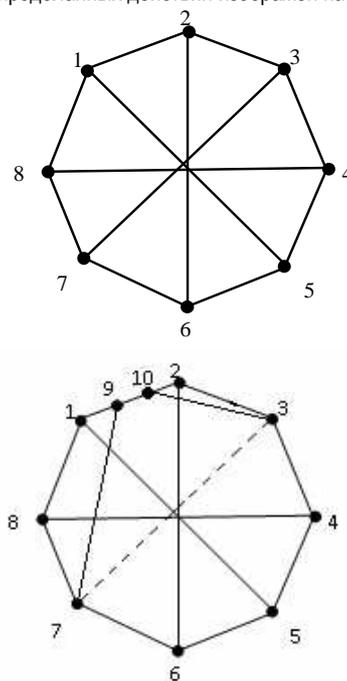


Рис. 3. Пример графа

Очевидно, что степень регулярного графа при таком методе генерации сохраняется.

**4. Алгоритм ломки ребер.** Также к точным алгоритмам, но использующим послойную генерацию относится алгоритм ломки ребер. Разработанный алгоритм основан на послойной генерации множества графов  $K_n^m$  из уже сгенерированного множества графов

$K_{n-2}^m$ . Общий принцип последовательной генерации на примере

множества графов  $K_8^3$  изображен на рисунке 2.

Для того чтобы сгенерировать все возможные комбинации ребер в контуре, будем строить графы, руководствуясь следующей последовательностью действий:

Шаг 1. Выбираем граф из множества  $K_{n-2}^m$ . Если необработанных графов больше нет – алгоритм закончен.

Шаг 2. Выбираем ведущее ребро в графе, исходя из следующих условий:

1. ребро является внешним, т.е. принадлежит кольцу;
2. ребро еще не было ведущим.

Если такого ребра нет, переходим к шагу 1.

Шаг 3. «Ломаем» ведущее ребро путем добавления между вершинами, которым оно инцидентно новой вершины  $n+1$ . Фиксируем эту вершину как одну из вершин будущего ребра нового графа.

Шаг 4. Выбираем конечное ребро, исходя из следующих условий:

1. ребро является внешним, т.е. принадлежит кольцу;
2. ребро еще не было ведущим;
3. ребро еще не ломалось одновременно с текущим ведущим ребром.

Если такого ребра нет, переходим к шагу 2.

Шаг 5. «Ломаем» конечное ребро, таким же образом как ведущее, путем добавления между вершинами, которым оно инцидентно новой вершины  $n+2$ .

Шаг 6. Соединяем вершины  $n+1$  (фиксированная) и  $n+2$ .

Шаг 7. Сохраняем граф. Переходим к шагу 4.

Для примера рассмотрим граф  $K_8^3$  изображенный на рисунке 4.

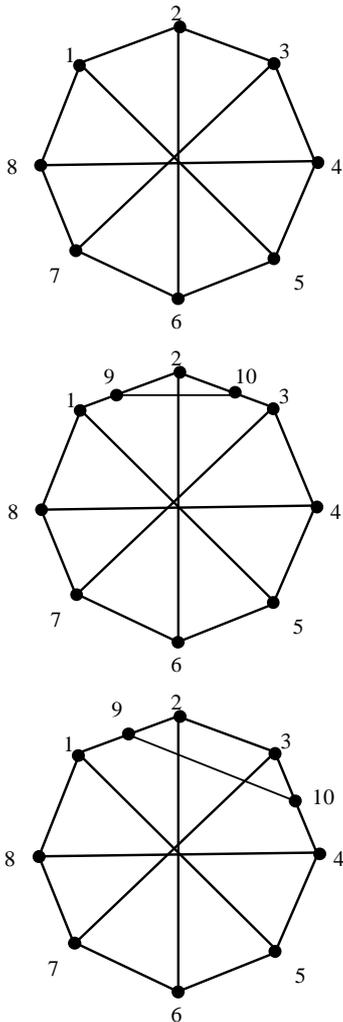


Рис. 4. Пример графа  $K_8^3$  и полученные графы  $K_{10}^3$

Очевидно, что степень регулярного графа при таком методе генерации сохраняется.

**5. Сравнительный анализ прямых алгоритмов генерации регулярных графов.** Теоретическое сравнение по трудоемкости для некоторых алгоритмов генерации приведено в таблице 1.

Таблица 1. Сравнение методов генерации регулярных графов

Название метода	Тип алгоритма	Трудоемкость	Память
Прямого произведения	точный	$n^2$	$n^2$
Алгоритм дерева	точный	$(n-2)!$	$n^3$
Генетический алгоритм	приближенный	$k \times n$	$k \times n^2$
Циклический перебор	точный	$n!$	$n^2$

Экспериментальный анализ методов проводился по среднему времени работы алгоритма.

Для наглядности изобразим данные, экспериментально полученные при генерации графов, на графике (рис. 5).

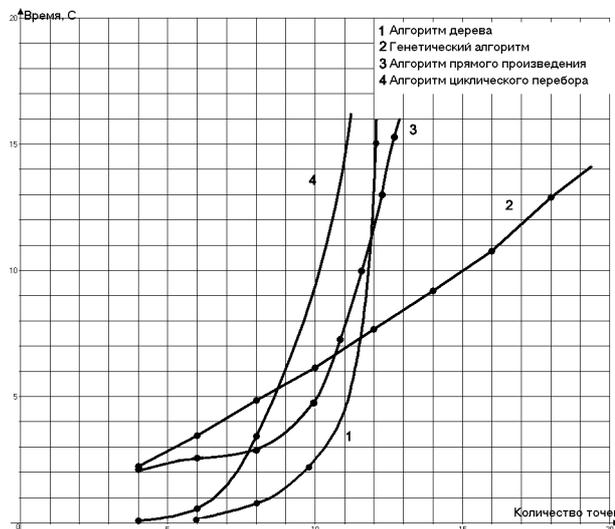


Рис. 5. Трудоемкость алгоритмов при генерации графов со степенью вершины

В результате экспериментов прямыми алгоритмами генерации регулярных графов для количества вершин от 4 до 24 и степеней вершин от 3 до 5 были получены результаты, из которых можно сделать вывод о том, что на малом количестве вершин классические (точные) алгоритмы выигрывают по скорости у генетического (неточный) алгоритма, и незначительно уступают ему по использованию памяти. Однако при увеличении числа вершин у классических алгоритмов значительно увеличивается время генерации. Т.о. приближенные алгоритмы незаменимы при необходимости генерации графов с большим количеством вершин. Тем не менее, иногда важнее получить все возможные графы, тогда предпочтительнее использовать точные алгоритмы. Помимо этого у всех прямых алгоритмов присутствует ряд прочих достоинств и недостатков.

**Выводы.** Существует достаточно много алгоритмов генерации однородных графов. Их можно разделить на точные и приближенные. А также на прямые и послойные. Прямые алгоритмы обладают рядом достоинств и недостатков. В работе был подробно описан один из таких алгоритмов – метод циклического перебора. К недостатком почти всех прямых алгоритмов можно отнести высокую трудоемкость, генерацию графов с большим избытком и прочее. Чтобы устранить или минимизировать эти недостатки были разработаны послойные алгоритмы. Послойные алгоритмы для создания нового

слова используют предыдущий. Одним из них является метод ломки ребер. Данный алгоритм показал хорошие результаты, и наряду с точностью, обладает также достаточно низкой трудоемкостью. Является простым для понимания и обеспечивает генерацию графов с любым заданным количеством вершин.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Gunnar Brinkmann: Fast Generation of Cubic Graphs. Journal of Graph Theory Vol. 23, No. 2 (1996), 139-149.
2. J.H. Kim, N.C. Wormald. Random matchings which induce Hamilton cycles, and Hamiltonian decompositions of random regular graphs. J. Combin. Theory Ser. B 81 (2001), p. 20-44.

3. M. Meringer. Erzeugung regulärer Graphen. Diplom arbeit bei Prof. Dr. Laue Lehrstuhl für Mathematik der Universität Bayreuth. Bayreuth, Januar, 1996.
4. Шуть В.Н., Свирский В.М., Муравьев Г.Л., Анфилец С.В. Генерация регулярных связных графов. Вестник БГТУ.– 2006. – № 5 (41).- С. 42-47.
5. Шуть В.Н., Иванюк Д.С., Свирский В.М., Теленкевич Р.С. Эвристические алгоритмы определения изоморфизма графов. Вестник БГТУ.– 2006. – № 5 (41).- С. 53-55.
6. Мелихов А.Н., Карелин В.П. Методы распознавания изоморфизма и изоморфного вложения четких и нечетких графов. - М.: Мир, 1995.

Материал поступил в редакцию 13.10.2008

#### SHUT V.N., VOJSENOVICH O.Y. Algorithms of level-by-level generation regular graph

In the work the problem of regular graphs generation is considered. One of methods of direct generation is a method of cyclic exhaustion, and one of methods of level-by-level generation is an algorithm of breaking edge. They are described in detail. The comparative analysis of direct algorithms is brought.

УДК 656.13.05

**Шуть В.Н., Партин В.С., Анфилец С.В., Касьяник В.В.**

## ОПТИМИЗАЦИЯ УПРАВЛЕНИЯ АВТОТРАНСПОРТОМ ПЕРЕД СВЕТОФОРОМ В УЛИЧНО-ДОРОЖНОЙ СЕТИ ГОРОДА

**Введение.** Бурный процесс автомобилизации с каждым годом охватывает всё большее число стран, постоянно увеличивается автомобильный парк, количество вовлекаемых в сферу дорожного движения людей. Рост автомобильного парка и объем перевозок ведёт к увеличению интенсивности движения, что в условиях городов с исторически сложившейся застройкой приводит к возникновению транспортной проблемы. Особенно остро она проявляется в узловых пунктах улично-дорожной сети. Здесь увеличиваются транспортные задержки, образуются очереди и заторы, что вызывает снижение скорости сообщения, неоправданный перерасход топлива и повышенное изнашивание узлов и агрегатов транспортных средств. Переменный режим движения, частые остановки и скопления автомобилей на перекрёстках являются причинами повышенного загрязнения воздушного бассейна города продуктами неполного сгорания топлива. Городское население постоянно подвержено воздействию транспортного шума и отработанных газов [1-3].

Организационные мероприятия способствуют упорядочению движения на уже существующей (сложившейся) улично-дорожной сети. К числу таких мероприятий относится организация одностороннего движения, кругового движения на перекрёстках, организация пешеходных переходов и пешеходных зон, автомобильных стоянок, остановок общественного транспорта и др.

В то время как организация мероприятий архитектурно-планировочного характера требует, помимо значительных капиталовложений, довольно большого периода времени, организационные мероприятия способны привести хотя и к временному, но сравнительно быстрому эффекту, в ряде случаев организационные мероприятия выступают в роли единственного средства для решения транспортной проблемы.

Речь идёт об организации движения в исторически сложившихся кварталах старых городов, которые часто являются памятниками архитектуры и не подлежат реконструкции. Кроме того, развитие улично-дорожной сети нередко связано с ликвидацией зеленых насаждений, что не всегда является целесообразным [4].

За последние годы за рубежом интенсивно ведутся работы по созданию автоматизированных систем с применением управляющих

ЭВМ, средств автоматизации, телемеханики, диспетчерской связи и телевидения для управления движением в масштабах крупного района или целого города [5,6]. Опыт эксплуатации таких систем убедительно свидетельствуют об их эффективности в решении транспортной проблемы.

**1. Постановка задачи.** В последние годы за рубежом находят широкое распространение адаптивные системы управления автотранспортом. На Западе разрабатываются системы управления четвертого поколения, учитывающие дорожную обстановку, интенсивность транспортных потоков, скорость автомобилей, фазы дорожного движения.

Фирма "Tune & Wear" (Великобритания) представила "умный" светофор, включающий мини-камеры, которые оценивают дорожную обстановку и устанавливают периодичность переключения цветов. Устройство идеально подходит для борьбы с дорожными пробками.

В Институте информации и системного анализа г.Мануа (Италия) создан светофор, оснащенный системой анализа загруженности дорог, который очень точно приспосабливается к различным ситуациям в городском дорожном движении, кроме того, он не нуждается в централизованном компьютерном управлении, поэтому обходится гораздо дешевле. Светофор сам меняет фазы зелёного и красного света в зависимости от плотности движения на перекрёстке.

Главное полицейское управление Японии (NPA) сообщило о создании нового типа светофора, который самостоятельно "разруливает" заторы. С помощью специальных датчиков фиксируется интенсивность движения на перекрёстке, и в случае возникновения «пробки» светофор автоматически выбирает наиболее подходящий для её рассасывания режим смены красного света на зелёный. Экспериментальная система подтвердила эффективность данного метода.

Власти Чикаго (США) предложили решение проблемы бесконечных пробок, которая в городе стоит крайне остро. На 2900 городских перекрёстках будут установлены камеры и датчики новой системы регулирования дорожного движения, связанные с «умными» светофорами. Свет автоматически будет переключаться в соответствии с ситуацией на перекрёстке.

**Партин Владимир Степанович**, заместитель директора по новой технике Брестского радиотехнического завода.

**Анфилец Сергей Викторович**, аспирант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

**Касьяник Валерий Викторович** – студент 5-го курса ФЭИС Брестского государственного технического университета. Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.