

УДК 519.17

ИНДЕКС ВИНЕРА МАКСИМАЛЬНЫХ ВНЕШНЕПЛОСКИХ ГРАФОВ

Ю. Л. Носов

г. Литецк, Россия

E-mail: yl.nosov@yandex.ru

Рассматривается инвариант $W(G)$ связных неориентированных графов G , равный сумме расстояний между всеми парами вершин графа G . Предлагается эффективный алгоритм расчёта матрицы расстояний и индекса Винера максимальных внешнеплоских графов с большим количеством вершин. Временная сложность алгоритма $O(n^2)$. Алгоритм удобен как для ручного расчёта индекса Винера небольших графов, так и для расчёта индекса Винера графов, сгенерированных компьютерной программой.

Ключевые слова: индекс Винера, максимальный внешнеплоский граф, хордальный граф, компактное представление хордального графа.

Введение

Рассматриваются конечные связные неориентированные графы $G(V, E)$ без петель и кратных рёбер с множеством вершин $V(G)$ и множеством рёбер $E(G)$. Числа вершин и рёбер графа обозначаются $n(G)$ и $m(G)$ соответственно (или для краткости n и m). Число вершин графа называется порядком графа, а число рёбер — его размером. Длина кратчайшей по числу рёбер цепи, соединяющей вершины u и v в графе G , называется расстоянием между вершинами u и v и обозначается через $d(u, v)$. Матрицей расстояний n -вершинного графа G называется квадратная матрица $D(G) = [d_{ij}]$ размера $n \times n$, каждый элемент d_{ij} которой равен расстоянию между вершинами i и j . Матрица расстояний $D(G)$ является симметричной и все её диагональные элементы равны нулю.

Все неопределяемые ниже термины можно найти в [1].

Индексом Винера $W(G)$ графа G называется инвариант, определяемый как сумма расстояний между всеми парами вершин графа G , то есть

$$W(G) = \sum_{\{u,v\} \subseteq V(G)} d(u, v) = \frac{1}{2} \sum_{v \in V(G)} \sum_{u \in V(G)} d(u, v).$$

В терминах матрицы расстояний индекс Винера есть полусумма всех элементов матрицы или сумма всех элементов матрицы, находящихся выше главной диагонали:

$$W(G) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_{ij} = \sum_{i=1}^n \sum_{j=i+1}^n d_{ij} = \sum_{j=1}^n \sum_{i=1}^j d_{ij}.$$

Впервые этот инвариант был использован американским химиком Г. Винером в 1947 г. для установления корреляционных зависимостей между значениями W и свойствами ациклических химических соединений, молекулярные графы которых являются деревьями [2]. Индекс Винера является топологическим инвариантом, отражающим структурные свойства графа (разветвлённость, цикличность и компактность) и используется в качестве структурного дескриптора в многочисленных приложениях

(см., например, [3]). Наибольшее применение индекс Винера нашел в химии, где он рассматривается как мера подобия молекулярных графов и используется для создания химических соединений с заданными свойствами [4–8].

Работы по тематике, связанной с индексом Винера, ведутся по двум направлениям:

- исследование теоретико-графовых свойств индекса Винера;
- разработка алгоритмов для расчёта индекса Винера.

Большинство работ по исследованию теоретико-графовых свойств индекса Винера посвящено поиску зависимостей значения индекса от структурных особенностей графа, определению верхних и нижних оценок индекса Винера по значениям других инвариантов [4, 9–11]. В некоторых работах исследуется изменение индекса при операциях над графами (удалении или добавлении вершин) [12, 13]. Так, например, в работе [13] определено значение индекса Винера дерева при добавлении новой вершины и одного ребра, соединяющего его с одной из вершин дерева.

Обширный перечень работ по данному направлению приведён в [9]. Там же дан обзор по индексу Винера для деревьев и графов гексагональных систем, приводятся результаты по вычислению индекса Винера для графов, обсуждаются влияние структуры графов на значения индекса Винера, проблемы вырождения индекса и смежные вопросы.

Теория индекса Винера для рёберных графов $L(G)$ развита в работах [4, 10, 11]. Интерес к этому виду производных графов обусловлен тем, что его характеристики используются для характеристики «сложности» структуры исходного графа. Здесь основное внимание уделяется нахождению графов, для которых их индексы Винера и индексы их рёберных графов совпадают.

Графы со свойством $W(G) = W(L(G))$ и заданными цикломатическим числом и обхватом построены в [10, 11].

В работах по второму направлению обычно определяются простые расчётные выражения, использующие основные структурные параметры графов; разрабатываются специализированные алгоритмы расчёта индекса Винера для различных классов графов, а также «универсальные» алгоритмы расчёта индекса Винера произвольных графов, использующие матрицу расстояний. Причём, как указано в [9], определённое значение имеет и разработка простых алгоритмов расчёта индекса Винера без использования компьютера.

Особенно большой прогресс сделан в разработке расчётных выражений [5] и специализированных алгоритмов расчёта индекса Винера для различных видов деревьев, дендримеров, графов гексагональных и пентагональных систем [4, 6–9, 13].

Индекс Винера произвольных графов вычисляется по матрице расстояний за время $O(n^2)$. Таким образом, временная сложность алгоритма расчёта индекса Винера определяется временной сложностью алгоритма расчёта матрицы расстояний, минимально возможное значение которой равно $O(n^2)$.

Расчёт матрицы расстояний можно выполнить с помощью алгоритмов Флойда — Уоршелла [14, 15], Джонсона [16] или алгоритма умножения матриц [17]. Матрицу расстояний можно получить также посредством n -кратного выполнения алгоритма, определяющего длины кратчайших путей между заданной вершиной и всеми другими вершинами, каждый раз выбирая в качестве исходной новую вершину графа. В этом случае можно воспользоваться алгоритмом Дейкстры [18], алгоритмом поиска в ширину (BFS) [1, 19] или алгоритмом Беллмана — Форда [20, 21].

Алгоритм Дейкстры имеет временную сложность $O(n^2 \log n + nm)$, алгоритм поиска в ширину — $O(n^2 + nm)$, а алгоритм Беллмана — Форда — $O(mn^2)$ (здесь имеется в виду временная сложность по расчёту длины кратчайших путей между всеми парами вершин).

Алгоритм умножения матриц имеет временную сложность $O(n^3 \log n)$, алгоритм Флойда — Уоршелла — $O(n^3)$, а алгоритм Джонсона — $O(nm + n^2 \log n)$. Алгоритм Джонсона использует в качестве подпрограмм алгоритмы Дейкстры и Беллмана — Форда. Алгоритм Беллмана — Форда применяется при наличии отрицательных весов рёбер. В случае только положительных весов рёбер используется алгоритм Дейкстры. Таким образом, в рассматриваемом случае, когда все веса рёбер равны 1, результаты работы алгоритмов Джонсона и Дейкстры совпадают.

Нетрудно заметить, что все известные в настоящее время алгоритмы расчёта матрицы расстояний произвольных графов являются полиномиальными, а их временные сложности распределены в узком диапазоне от $O(n^3 \log n)$ до $O(n^2 + nm)$. Поэтому задача поиска быстрых алгоритмов расчёта индекса Винера графов очень большого порядка является актуальной.

Интересным для приложений классом графов являются максимальные внешне-плоские графы — плоские графы, у которых все вершины принадлежат внешней грани, а все внутренние грани есть треугольники [1, 22]. Графы этого вида входят в класс хордальных графов и представляют собой триангуляции выпуклого многоугольника, применяемые в системах искусственного интеллекта для обнаружения фигур в изображениях [23].

Индекс Винера максимальных внешне-плоских графов, отражающий их структуру и разветвлённость, может быть использован для распознавания и классификации графических образов в изображениях. Таким образом, задача разработки быстрого алгоритма расчёта индекса Винера максимальных внешне-плоских графов большого порядка является актуальной.

В настоящей работе предлагается простой и быстрый алгоритм расчёта матрицы расстояний и индекса Винера максимальных внешне-плоских графов большого порядка с временной сложностью $O(n^2)$.

1. Структура максимальных внешне-плоских графов

Рассмотрим структуру и свойства максимальных внешне-плоских графов (МВП-графов) [1], которые будут использованы для разработки алгоритма расчёта индекса Винера. Сначала отметим ряд основных хорошо известных свойств (см., например, [22, 24, 25]).

Утверждение 1 [22]. Любой МВП-граф G с $n \geq 3$ вершинами имеет:

- 1) $(n - 2)$ внутренних граней (каждая внутренняя грань — треугольник);
- 2) $(2n - 3)$ рёбер, из которых $(n - 3)$ — внутренние рёбра;
- 3) по крайней мере две вершины степени 2.

Утверждение 2 [25]. Из любого n -вершинного МВП-графа можно получить другой МВП-граф с $(n + 1)$ вершинами посредством добавления одной вершины и двух рёбер, соединяющих эту вершину с двумя вершинами ребра, принадлежащего внешней грани исходного МВП-графа.

Утверждение 3. При удалении вершины степени 2 и двух инцидентных ей рёбер из n -вершинного МВП-графа образуется новый $(n - 1)$ -вершинный МВП-граф.

МВП-графы являются хордальными графами [24] и, следовательно, имеют не менее двух симплициальных вершин (вершин, у которых множество смежных вершин индуцирует полный подграф). В МВП-графах симплициальные вершины имеют степень 2.

По количеству симплициальных вершин все МВП-графы можно разделить на два класса. МВП-графы первого класса имеют точно две симплициальные вершины. МВП-графами второго класса являются графы, имеющие не менее трёх симплициальных вершин. В качестве примера на рис. 1, *а–в* представлены МВП-графы первого класса, а на рис. 1, *г* — МВП-граф второго класса. Пунктиром показаны рёбра внешней грани, незакрашенными кружками — симплициальные вершины.

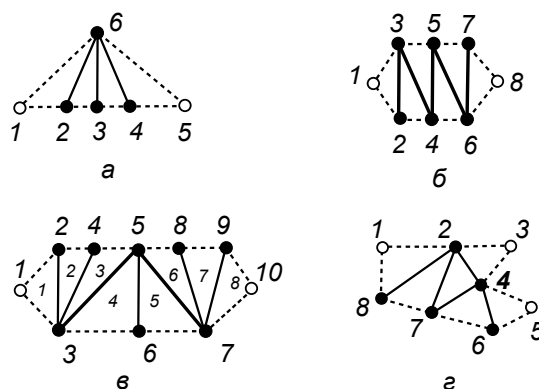


Рис. 1. МВП-графы первого класса: *а* — типа «веер»,
б — «лестница», *в* — «цепь»; *г* — второго класса

Из свойства хордальности МВП-графов следует (см., например, [24]) справедливость следующего утверждения.

Утверждение 4. Любой n -вершинный МВП-граф имеет совершенную последовательность исключения [1], т. е. последовательность вершин v_1, v_2, \dots, v_n , в которой любая вершина v_i , $i = 1, \dots, n - 1$, является симплициальной вершиной в подграфе G_i , индуцированном множеством вершин $\{v_i, \dots, v_n\}$.

Совершенная последовательность исключения МВП-графа может быть получена за линейное время с помощью алгоритмов LexBFS или MCS [26, 27].

Операция добавления вершины и двух рёбер, в литературе иногда называемая элементарным расширением [25], является основой для рекурсивной характеристики МВП-графов [25, 28].

Утверждение 5. Граф $G \neq K_3$ является МВП-графом тогда и только тогда, когда он может быть получен из графа K_3 с помощью конечной последовательности элементарных расширений.

Из утверждения 5 следует, что любой n -вершинный МВП-граф G_n получается из графа K_3 в результате последовательного k -кратного ($k = n - 3$) выполнения операции элементарного расширения.

Возможность получения МВП-графа посредством конечной последовательности элементарных расширений определяет один из возможных способов нумерации вершин МВП-графа, который будем называть нумерацией в порядке построения (или естественной нумерацией).

Нумерацией в порядке построения (или естественной нумерацией) вершин n -вершинного МВП-графа называется такая нумерация, при которой для любого $3 \leq i \leq n$ подграф $G_i = G[\{1, 2, 3, \dots, i\}]$, индуцированный множеством вершин $\{1, 2, 3, \dots, i\}$, является МВП-графом.

В качестве примера на рис. 2 представлен процесс нумерации вершин МВП-графа в порядке построения.

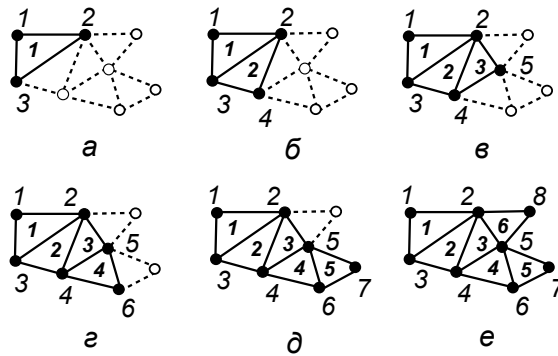


Рис. 2. Процесс нумерации вершин МВП-графа в порядке построения

Такая нумерация возникает при построении n -вершинного МВП-графа из K_3 посредством k -кратного выполнения элементарного расширения. Для этого необходимо сначала пронумеровать вершины исходного графа K_3 числами 1, 2, 3, а все остальные вершины нумеровать последовательно в порядке их добавления в граф. МВП-графы с нумерацией в порядке построения могут быть созданы с помощью компьютерной программы.

Примеры естественной нумерации в порядке построения представлены на рис. 1, б, в. Нумерация в порядке построения отображает структуру МВП-графа, поэтому будем её использовать для анализа метрических свойств МВП-графов и для разработки алгоритма расчёта матрицы расстояний и индекса Винера.

2. Компактное представление МВП-графов

В работе [29] для хордального графа G , имеющего симплициальную декомпозицию [24] со свойством скользящего пересечения (running intersection property) [30], предложено компактное представление $CP[G]$ в виде такой последовательности пар

$$CP(G) = [(R_1, S_1), (R_2, S_2), \dots, (R_p, S_p)], \quad (1)$$

что

$$R_1 = Q_1, \quad S_1 = \emptyset, \quad R_j = Q_j \setminus S_j, \quad S_j = Q_j \cap (Q_1 \cup Q_2 \cup \dots \cup Q_{j-1}), \quad 2 \leq j \leq p,$$

где Q_i — i -я максимальная клика; S_i — i -е минимальное разделяющее множество (сепаратор); R_i — i -е множество остатков, состоящее из симплициальных вершин.

В [29] показано, что последовательность R_p, R_{p-1}, \dots, R_1 является совершенной последовательностью исключения.

В МВП-графах максимальная клика Q_1 и каждая максимальная клика Q_j состоит из трёх вершин, каждое множество S_j состоит из двух вершин, а каждое множество R_j имеет только одну симплициальную вершину.

Тогда, подставляя в формулу (1) значения составляющих Q_1, Q_j, S_j и $p = n - 2$,

$$Q_1 = \{v'_1, v'_2, v'_3\}, \quad R_j = v_j, \quad Q_j = S_j \cup R_j, \quad S_j = \{x_j, y_j\}, \quad 2 \leq j \leq n - 2,$$

для компактного представления МВП-графа G получим

$$CP(G) = [(\{v'_1, v'_2, v'_3\}, \emptyset), (v_2, \{x_2, y_2\}), \dots, (v_{n-2}, \{x_{n-2}, y_{n-2}\})], \quad (2)$$

где v'_1, v'_2, v'_3 — симплициальные вершины, входящие в клику Q_1 ; v_j — симплициальная вершина клики Q_j ; x_j, y_j — вершины из множества $(Q_1 \cup Q_2 \cup \dots \cup Q_{j-1})$, смежные с вершиной v_j .

Таким образом, компактное представление МВП-графа G — это последовательность троек вершин, образующих максимальные клики G .

Полученное выражение (2) для n -вершинного МВП-графа G можно представить как результат последовательного применения операции элементарного расширения.

Действительно, пусть $G_1, G_2, \dots, G_i, \dots, G_{n-2}$ — последовательность графов, полученных в результате последовательного применения операции элементарного расширения. Тогда для этой последовательности справедливы выражения

$$G_1 = Q_1 = K_3, \quad G_j = Q_1 \cup Q_2 \cup \dots \cup Q_j, \quad 2 \leq j \leq n - 2,$$

и каждый граф G_j получается из графа G_{j-1} добавлением новой вершины v_j и двух рёбер, соединяющих её с парой вершин x_{j-1}, y_{j-1} графа G_{j-1} .

Очевидно, что для МВП-графа с нумерацией вершин в порядке построения существует компактное представление

$$CP(G) = [(\{1, 2, 3\}, \emptyset), (4, \{x_2, y_2\}), (5, \{x_3, y_3\}), \dots, (n, \{x_{n-2}, y_{n-2}\})]. \quad (3)$$

Компактное представление МВП-графа с нумерацией вершин в порядке построения вида (3) будем называть *каноническим компактным представлением* и обозначать через $CCP(G)$.

Пример 1. МВП-граф на рис. 1, б имеет каноническое компактное представление

$$CCP(G) = [(\{1, 2, 3\}, \emptyset), (4, \{2, 3\}), (5, \{3, 4\}), (6, \{4, 5\}), (7, \{5, 6\}), (8, \{6, 7\})].$$

Пример 2. МВП-граф на рис. 1, г имеет компактное представление

$$CP(G) = [(\{1, 2, 8\}, \emptyset), (7, \{2, 8\}), (4, \{2, 7\}), (3, \{2, 4\}), (6, \{4, 7\}), (5, \{4, 6\})].$$

В [29] показано, что компактное представление МВП-графа может быть получено из его совершенной последовательности исключения за время $O(n + m)$. Однако более целесообразно использовать алгоритм, предложенный в [31]. Алгоритм использует метод MCS и имеет временную сложность $O(n + m)$.

Каноническое компактное представление МВП-графа G получается из компактного представления $CP(G)$ посредством перенумерации вершин. Псевдокод алгоритма приводится ниже.

Алгоритм 1. Алгоритм расчёта $CCP(G)$ МВП-графа G **Вход:** компактное представление $CP(G)$ **Выход:** каноническое компактное представление $CCP(G)$ 1: $index[v'_1] := 1; index[v'_2] := 2; index[v'_3] := 3;$ 2: $v'_1 := 1; v'_2 := 2; v'_3 := 3;$ 3: **Для** $j = 2, \dots, n - 2$ 4: $index[v_j] := j + 2; v_j := j + 2;$ 5: **Для** $j = 2, \dots, n - 2$ 6: $x_j := index[x_j]; y_j := index[y_j].$

Алгоритм перенумерации достаточно прост и состоит из трёх шагов. На первом шаге создается индексный массив $index$ длины n . Затем для каждой из вершин v'_1, v'_2, v'_3 определяется индекс $index[v'_1] = 1, index[v'_2] = 2, index[v'_3] = 3$, после чего номерам вершин v'_1, v'_2, v'_3 присваиваются новые значения $v'_1 = 1, v'_2 = 2, v'_3 = 3$.

На втором шаге в цикле выполняется проход по всем другим триплетам $\{v_2, \{x_2, y_2\}\}, \dots, \{v_{n-2}, \{x_{n-2}, y_{n-2}\}\}$ и для каждой вершины $v_j, j = 2, \dots, n - 2$, определяется индекс $index[v_j] = j + 2$, а номеру вершины присваивается новое значение $v_j = j + 2$. Количество повторений в цикле равно $n - 3$.

На третьем шаге в цикле выполняется повторный проход по триплетам и в каждой паре вершин $\{x_j, y_j\}, j = 2, \dots, n - 2$, вершинам x_j и y_j присваиваются новые значения номеров с помощью индексного массива по формулам $x_j = index[x_j]$ и $y_j = index[y_j]$. Таким образом, при перенумерации выполняются два цикла, в каждом из которых количество повторений равно $n - 3$. Временная сложность алгоритма перенумерации составляет $O(n)$.

Отметим, что каноническое компактное представление МВП-графа может быть сгенерировано компьютерной программой в процессе создания МВП-графа.

3. Метрические свойства МВП-графов

Операция элементарного расширения (добавления вершины и двух рёбер) является очень важным звеном в построении МВП-графов. Поэтому вопрос, как изменяется индекс Винера МВП-графа при выполнении этой операции, имеет важное значение.

Теорема 1. Пусть G_n — МВП-граф с $n > 3$ вершинами $v_1, v_2, v_3, \dots, v_n$ и индексом Винера $W(G_n)$. Тогда МВП-граф G_{n+1} , полученный добавлением вершины v_{n+1} и двух рёбер, соединяющих её с вершинами v_p и v_q ребра, принадлежащего внешней грани МВП-графа G_n , имеет индекс Винера, равный

$$W(G_{n+1}) = W(G_n) + \sum_{i=1}^n (\min(d(v_i, v_p), d(v_i, v_q)) + 1). \quad (4)$$

Доказательство. По определению индекс Винера МВП-графа G_{n+1} равен

$$W(G_{n+1}) = \sum_{i=1}^n \sum_{j=i+1}^{n+1} d(v_i, v_j).$$

Правую часть этого выражения можно представить в виде

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n d(v_i, v_j) + \sum_{i=1}^n d(v_i, v_{n+1}). \quad (5)$$

Очевидно, первый член выражения (5) есть индекс Винера исходного МВП-графа, а для любой вершины $v_i \in V(G_n)$ расстояние $d(v_i, v_{n+1})$ между вершинами v_i и v_{n+1} равно

$$d(v_i, v_{n+1}) = \min(d(v_i, v_p), d(v_i, v_q)) + 1. \quad (6)$$

Подставляя значение $d(v_i, v_{n+1})$ из равенства (6) в выражение (5), получим искомое выражение (4). ■

Выражения (4), (6) могут быть использованы в алгоритме расчёта индекса Винера МВП-графов.

4. Расчет индекса Винера МВП-графов

Нетрудно заметить, что для МВП-графов с естественной нумерацией вершин можно построить простой итеративный алгоритм для определения матрицы расстояний и расчёта индекса Винера. Действительно, пусть G_n — n -вершинный МВП-граф с естественной нумерацией, полученный из исходного трёхвершинного МВП-графа K_3 последовательным добавлением $n - 3$ вершин. В процессе построения графа получим последовательность его подграфов вида

$$G_3 = K_3, G_4, G_5, G_6, \dots, G_{n-1}, G_n.$$

Тогда для индекса Винера $W(G_k)$ каждого подграфа G_k будем иметь

$$W(G_k) = W(G_{k-1}) + S_k, \quad k = 4, 5, \dots, n - 1, n,$$

где S_k — приращение индекса Винера при добавлении вершины k .

В результате для индекса Винера МВП-графа G_n получим

$$W(G_n) = W(G_3) + S_4 + S_5 + \dots + S_{n-1} + S_n. \quad (7)$$

Значение приращения индекса Винера для k -й вершины (при её добавлении) получим из выражения (3), используя матричную нотацию

$$S_k = \sum_{i=1}^{k-1} (\min(d_{ip}, d_{iq}) + 1), \quad (8)$$

где p и q — вершины подграфа G_{k-1} , смежные с вершиной k подграфа G_k ; d_{ip}, d_{iq} — элементы матрицы расстояний $[d_{ij}]$.

На основании (8) предлагается алгоритм 2 расчёта матрицы расстояний и индекса Винера МВП-графа, заданного в каноническом компактном представлении. Основными исходными данными в алгоритме являются два массива, содержащие номера вершин, смежных с добавляемыми вершинами, и матрица расстояний, у которой на первой стадии расчётов инициализируются диагональные элементы и элементы подматрицы размера 3×3 , соответствующей исходному трёхвершинному МВП-графу.

Расчёт выполняется последовательно для всех подграфов G_k , начиная с $k = 4$, при этом сначала определяются элементы верхней части k -го столбца подматрицы размера $k \times k$, а затем — элементы левой части k -й строки.

Докажем корректность алгоритма. Ясно, что для правильной работы алгоритма необходимо выполнение следующего условия: на каждом шаге алгоритма при добавлении вершины k к графу G_{k-1} расстояния $d(v_i, v_p)$ и $d(v_i, v_q)$ от любой вершины $v_i \in G_{k-1}$ до вершин v_p и v_q , смежных с вершиной k , должны быть определены на предшествующих шагах алгоритма.

Алгоритм 2. Алгоритм расчета матрицы расстояний и индекса Винера**Вход:** целочисленные массивы номеров вершин $A[n]$ и $B[n]$ размера n **Выход:** матрица расстояний $d[i, j]$, индекс Винера W

```

1: Для  $i = 1, \dots, n$ 
2:    $d[i, i] := 0$ ; // инициализация
3:  $d[1, 2] := 1; d[1, 3] := 1; d[2, 1] := 1$ ;
4:  $d[2, 3] := 1; d[3, 1] := 1; d[3, 2] := 1; W := 3$ ;
5: Для  $k = 4, \dots, n$ 
6:    $p := A[k]; q := B[k]; S[k] := 0$ ;
   // Заполнение верхней части  $k$ -го столбца матрицы расстояний
7:   Для  $i = 1, \dots, k - 1$ 
8:      $d[i, k] := \min(d[i, p], d[i, q]) + 1$ ;
9:      $S[k] := S[k] + d[i, k]$ ;
   // Заполнение левой части  $k$ -й строки матрицы расстояний
10:  Для  $j = 1, \dots, k - 1$ 
11:     $d[k, j] := d[j, k]$ ;
12:   $W := W + S[k]$  // Расчёт индекса Винера

```

Выполнение этого условия обеспечивается:

- 1) структурой алгоритма (на каждом шаге сначала производится расчёт всех элементов верхней части k -го столбца, которые затем копируются в соответствующие элементы левой части k -й строки, которая будет использована на следующем шаге);
- 2) использованием канонического компактного представления (следовательно, для любой вершины v_i графа G_{k-1} , к которому добавляется вершина k , выполняется неравенство $v_i < k$).

Определим временную сложность алгоритма. Алгоритм состоит из четырёх циклов. В первом цикле (операторы 1, 2) выполняется заполнение главной диагонали матрицы нулями. Все основные действия алгоритма выполняются во внешнем цикле (операторы 5–12) и двух внутренних циклах (операторы 7–9 и 10, 11). В первом внутреннем цикле производится расчёт всех элементов верхней части k -го столбца. Во втором внутреннем цикле производится заполнение всех элементов левой части k -й строки. Количество повторений в первом цикле n , в других циклах — $n^2 - n - 6$. Таким образом, временная сложность алгоритма $O(n^2)$.

Общая временная сложность расчёта индекса Винера по предлагаемой методике складывается из временной сложности алгоритма расчёта индекса Винера и алгоритмов получения компактного представления и его перенумерации. Сложность алгоритма получения компактного представления равна $O(n+m)$, алгоритма перенумерации — $O(n)$. Для МВП-графов $m = 2n - 3$, поэтому общее количество операций алгоритма равно $n^2 + 5n - 9$, а общая временная сложность — $O(n^2)$.

Ясно, что предлагаемый алгоритм превосходит по эффективности алгоритм умножения матриц, алгоритм Флойда — Уоршелла и алгоритмы Джонсона и Дейкстры, которые имеют временную сложность $O(n^3 \log n)$, $O(n^3)$ и $O(n^2 \log n + nm)$ соответственно. Алгоритм Беллмана — Форда при расчёте индекса Винера МВП-графов имеет временную сложность $O(n^3)$ (при n^3 имеется скрытая константа 2) и тоже уступает предлагаемому алгоритму по эффективности.

Из всех известных алгоритмов лучшим является алгоритм поиска в ширину (BFS), который при расчёте индекса Винера МВП-графов имеет временную сложность $O(n^2)$. Однако алгоритм BFS хуже, чем предлагаемый алгоритм, поскольку количество операций алгоритма BFS равно $(3n^2 - 3n)$, что при больших значениях n приблизительно в 3 раза больше общего количества операций предлагаемого алгоритма.

Это подтверждается результатами тестирования алгоритмов получения компактного представления (CP), расчёта индекса Винера по предлагаемой методике (W_{CP}) и с помощью алгоритма BFS (W_{BFS}). В таблице представлены средние значения времени работы алгоритмов в миллисекундах для разных значений количества вершин n . Расчёты проводились в системе *Mathematica*. Для каждого значения n с помощью компьютера генерировались случайные МВП-графы с нумерацией в порядке построения. Алгоритм BFS реализован в соответствии с [19]. Для получения компактного представления использовался алгоритм [31]. Нетрудно заметить, что при любом $n \geq 50$ время работы алгоритма BFS в 3 раза больше, чем суммарное время работы алгоритмов получения компактного представления и расчёта индекса Винера по предлагаемой методике.

n	CP	W_{CP}	W_{BFS}
50	16	22	123
75	31	47	275
100	48	89	508
150	98	211	1158

Выводы

Для МВП-графов получено рекуррентное соотношение для вычисления индекса Винера. Разработан алгоритм расчёта индекса Винера МВП-графов, представленных в компактной форме. Алгоритм имеет минимально возможное для алгоритмов, использующих матрицу расстояний, значение временной сложности равно $O(n^2)$. Наибольшая сравнительная эффективность достигается при расчёте индекса Винера МВП-графов большого порядка. Однако простота и наглядность алгоритма позволяют использовать его и для ручных вычислений индекса Винера МВП-графов с малым количеством вершин. Дополнительным положительным свойством алгоритма является возможность его прямого (без перенумерации вершин) использования для расчёта индекса Винера МВП-графов, сгенерированных компьютерной программой.

ЛИТЕРАТУРА

1. Евстигнеев В. А., Касьянов В. Н. Словарь по графам в информатике. Новосибирск: ООО «Сибирское научное издательство», 2009. 300 с.
2. Wiener H. Structural determination of paraffin boiling points // J. Amer. Chem. Soc. 1947. V. 69. P. 17–20.
3. Федянин Д. Н. Об индексе Винера в последовательности социальных сетей // Тр. 52-й науч. конф. МФТИ «Современные проблемы фундаментальных и прикладных наук». Ч. I. Радиотехника и кибернетика. Т. 2. М.: МФТИ, 2009. С. 94–97.
4. Dobrynin A. A., Entringer R. C., and Gutman I. Wiener index of trees: theory and applications // Acta Appl. Math. 2001. V. 66. No. 3. P. 211–249.
5. Gutman I., Yeh Y. N., Lee S. L., and Luo Y. L. Some recent results in the theory of the Wiener number // Indian J. Chemistry. 1993. V. 32A P. 651–661.
6. Gutman I., Yan W., Yang B. Y., and Yeh Y. N. Generalized Wiener indices of zigzagging pentachains // J. Math. Chem. 2007. V. 42. No. 2. P. 103–117.

7. *Gutman I., Yeh Y. N., Lee S. L., and Chen J. C.* Wiener numbers of dendrimer // *Commun. Math. Chem.* 1984. V. 30. No. 1. P. 103–115.
8. *Vukičević D. and Trinajstić N.* Wiener indices of benzenoid graphs // *Bulletin of the Chemists and Technologists of Macedonia.* 2004. V. 23. No. 2. P. 113–129.
9. *Добрынин А. А., Гутман И.* Индекс Винера для деревьев и графов гексагональных систем // *Дискретный анализ и исследование операций.* 1998. Сер. 2. Т. 5. № 2. С. 34–60.
10. *Добрынин А. А., Мельников Л. С.* Индекс Винера для графов и их реберных графов // *Дискретный анализ и исследование операций.* 2004. Сер. 2. Т. 11. № 2. С. 25–44.
11. *Добрынин А. А.* Индекс Винера для графов произвольного обхвата и их реберных графов // *Сибирский журнал индустриальной математики.* 2009. Т. XXII. № 4(40). С. 44–50.
12. *Šoltés L.* Transmission in graphs: a bound and vertex removing // *Math. Slovaca.* 1991. V. 41. No. 1. P. 11–16.
13. *El Marraki M. and Al Hagri G.* Calculation of the Wiener index for some particular trees // *J. Theor. Appl. Inform. Technology (JATIT).* 2010. V. 22. No. 2. P. 77–83.
14. *Floyd R. W.* Algorithm 97: Shortest Path // *Comm. ACM.* 1962. V. 5. No. 6. P. 345.
15. *Warshall S.* A theorem on Boolean matrices // *J. ACM.* 1962. V. 9. No. 1. P. 11–12.
16. *Johnson D. B.* Efficient algorithms for shortest paths in sparse networks // *J. ACM.* 1977. V. 24. No. 1. P. 1–13.
17. *Müller W. R., Szymanski K., Knop J. V., and Trinajstić N.* An algorithm for construction of the molecular distance matrix // *J. Comput. Chem.* 1987. V. 8. No. 2. P. 170–173.
18. *Dijkstra E. W.* A note on two problems in connexion with graphs // *Numer. Math.* 1959. V. 1. No. 1. P. 269–271.
19. *Mohar B. and Pisanski T.* How to compute the Wiener index of a graph. // *J. Math. Chem.* 1988. V. 3. No. 2. P. 267–277.
20. *Bellman R.* On a routing problem // *Quart. Appl. Math.* 1958. V. 16. No. 1. P. 87–90.
21. *Ford Jr. L. R.* Network Flow Theory // Paper P-923. Santa Monica, California: RAND Corp., 1956. P. 923.
22. *Харари Ф.* Теория графов. М.: Мир, 1973. 300 с.
23. *Felzenszwalb P. F.* Representation and detection of deformable shapes // *IEEE Trans. Pattern Analys. Machine Intelligence.* 2005. V. 27. No. 2. P. 208–220.
24. *Dirac G. A.* On rigid circuit graphs // *Abh. Math. Sem. Univ. Hamburg.* 1967. V. 25. P. 71–76.
25. *Asratian A. S. and Oksimets N.* Graphs with hamiltonian balls // *Australasian J. Combinator.* 1998. V. 17. No. 4. P. 185–198.
26. *Rose D., Tarjan R. E., and Lueker G.* Algorithmic aspects of vertex elimination on graphs // *SIAM J. Comput.* 1976. V. 5. No. 2. P. 146–160.
27. *Tarjan R. E., and Yannakakis M.* Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs // *SIAM J. Comput.* 1984. V. 13. No. 3. P. 566–579.
28. *Beyer T., Jones W., and Mitchell S.* Linear algorithms for isomorphism of Maximal outer-planar graphs. // *J. ACM.* 1979. V. 26. No. 4. P. 603–610.
29. *Markenzon L. and Pereira P. R. C.* A compact representation for chordal graphs // *Proc. of Workshop on Graphs and Combinatorial Optimization.* Gargnano: CTW, 2008. P. 174–176.
30. *Blair J. R. S. and Peyton B.* An introduction to chordal graphs and clique trees // *Graph Theory and Sparse Matrix Computation.* N.Y.: Springer Verlag, 1993. P. 1–29.
31. *Markenzon L. and Vernet O.* Representações Computacionais de Grafos. *Notas em Matemática Aplicada.* V. 24. São Carlos, SP: SBMAC, 2006. 76 p.