

Метод выделения изоморфного подграфа графа СЭП в САПР электронных схем

С.В. Курапов¹, М.В. Давидовский²

¹Запорожский национальный университет, г. Запорожье, Украина, lili5050@rambler.ru

²Запорожский институт последипломного педагогического образования, г. Запорожье, Украина
m.davidovsky@gmail.com

Аннотация — Актуальной задачей при создании современных САПР электронной аппаратуры является задача распознавания структур различных объектов. Данная работа представляет метод решения задачи выделения подграфа G_0 графа $G(V,E)$ изоморфного заданному эталонному графу $H(X,U)$. Метод основан на применении полного инварианта графа, описывающего его индивидуальную структуру в числовом представлении. Предварительное определение полного инварианта эталонного графа позволяет свести задачу к поиску множества подграфов графа G с количеством вершин равным мощности множества вершин $\text{card}(X)$ графа H . Решение сводится к перечислению всех множеств сочетаний из $|V|$ по $|X|$. Такая постановка задачи позволяет свести труднорешаемую задачу к вычислительным методам, имеющим полиномиальную вычислительную сложность.

Ключевые слова — система автоматизированного проектирования, граф схемы электрической принципиальной, изоморфизм графов, выделение изоморфного подграфа.

ВВЕДЕНИЕ

В состав современных САПР включаются программные инструменты реализующие математические методы распознавания структур различных объектов. Под методами распознавания структур графа будем понимать следующее: определение изоморфизма двух графов, определение группы автоморфизма графа, определение бинарного соответствия двух графов, выделение подграфа G_0 графа G изоморфного заданному графу H . Так на практике возникает необходимость проверки изоморфизма графов (например, для верификации различных представлений электронной схемы при автоматизации проектирования электронных схем) с целью повышения эффективности САПР электронной аппаратуры [1]. Традиционно данный круг задач принято относить к труднорешаемым задачам [2]. Созданием методов и эффективных алгоритмов решения задач данного круга занимаются многие исследователи [3,6-8,10-13]. В настоящее время накоплен большой арсенал методов для решения задач распознавания структур графа. В большинстве случаев методы с полиномиальной вычислительной сложностью носят эвристический характер, в то время как методы, основанные на точном решении, имеют экспоненциальную вычислительную сложность.

В основу предлагаемого метода решения задач данного круга положим полный инвариант графа, построенный на его центральных разрезах и являющийся индивидуальной характеристикой графа. То есть полный инвариант графа позволяет различать структуры графа аналогично методам распознавания личности с помощью папиллярных линий.

Рассмотрим состав полного инварианта графа G [9]:

- матрица инцидентности графа $B(G)$;
- n – количество вершин графа G ;
- m – количество ребер графа G ;
- k – количество уровней в спектре реберных разрезов графа G ;
- $\{\xi(w(r_1)), \xi(w(r_2)), \dots, \xi(w(r_k))\}$ – множество кортежей весов уровней;
- $\xi_e(G)$ – суммарный кортеж весов ребер;
- $\zeta_v(G)$ – кортеж весов вершин.
- $F_e(G) \& F_v(G)$ – базовый инвариант графа.

При сравнении кортежей удобно располагать элементы в порядке неубывания их величин, не учитывая их местоположения. В качестве примера рассмотрим сильно регулярные графы Lattice graph с параметрами $\text{srg}(16,6,2,2)$ и Shrikhande graph с параметрами $\text{srg}(16,6,2,2)$. Количество уровней в спектре реберных разрезов графа $G_1 = 2$. Кортеж весов ребер уровня 1: $\xi(w(r_1)) = \langle 48 \times 10 \rangle$. Кортеж весов ребер уровня 2: $\xi(w(r_2)) = \langle 48 \times 16 \rangle$. Суммарный кортеж весов ребер графа G_1 : $f_e(G_1) = \langle 48 \times 26 \rangle$. Кортеж весов вершин графа G_1 : $f_v(G_1) = \langle 16 \times 156 \rangle$. Базовый инвариант графа $F_e(G_1) \& F_v(G_1) = (48 \times 26) \& (16 \times 156)$.

Количество уровней в спектре реберных разрезов графа $G_2 = 2$. Кортеж весов ребер уровня 1: $\xi(w(r_1)) = \langle 48 \times 10 \rangle$. Кортеж весов ребер уровня 2: $\xi(w(r_2)) = \langle 48 \times 24 \rangle$. Суммарный кортеж весов ребер графа G_2 : $f_e(G_2) = \langle 48 \times 34 \rangle$. Кортеж весов вершин

графа G_2 : $f_v(G_2) = \langle 16 \times 204 \rangle$. Базовый инвариант графа $F_e(G_2) \& F_v(G_2) = (48 \times 26) \& (16 \times 204)$.

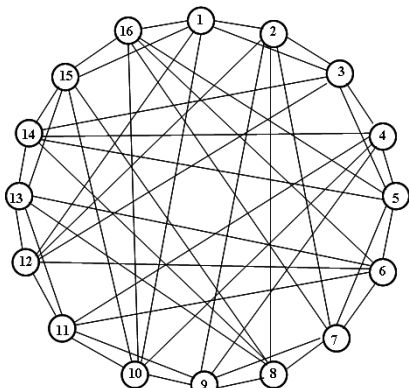


Рис. 1. Сильно регулярный граф G_1 (Lattice graph) с параметрами $\text{srg}(16,6,2,2)$

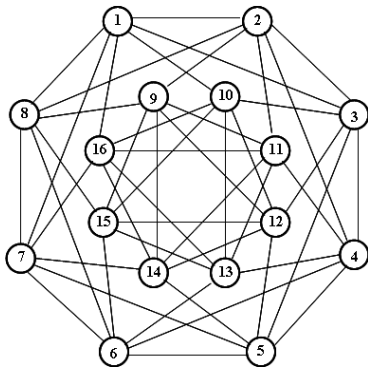


Рис. 2. Сильно регулярный граф G_2 (Shrikhande graph) с параметрами $\text{srg}(16,6,2,2)$

Сильно регулярные графы G_1 и G_2 не изоморфны, так как их базовые инварианты не совпадают. Рассмотрим в качестве примера, следующую пару изоспектральных графов (рис. 3 и рис. 4).

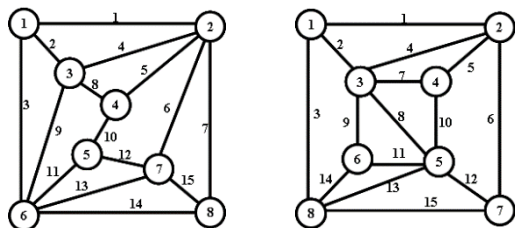


Рис. 3. Изоспектральные графы G_3 и G_4

Количество уровней в спектре реберных разрезов графа $G_3 = 6$. Кортежи весов ребер:

- уровня 1: $\xi(w(r_1)) = \langle 6,5,7,7,6, 7,6,5,7,4, 6,5,7,6,5 \rangle$;
- уровня 2: $\xi(w(r_2)) = \langle 9,5,7,8,9, 6,7,7,6,10, 9,7,8,9,5 \rangle$;
- уровня 3: $\xi(w(r_3)) = \langle 8,7,4,7,5, 7,4,6,7,8, 5,6,7,8,7 \rangle$;
- уровня 4: $\xi(w(r_4)) = \langle 10,6,10,4, 10,4,10,10, 4,0,10,10, 4,10,6 \rangle$;
- уровня 5: $\xi(w(r_5)) = \langle 8,10,8,10, 10,10,8,8, 10,0,10,8, 10,8,10 \rangle$;
- уровня 6: $\xi(w(r_6)) = \langle 0,8,0,8,8, 8,0,0,8,0, 8,0,8,0,8 \rangle$.

Суммарный кортеж весов ребер графа G_3 : $f_e(G_3) = \langle 41,41,35,4 4,48,42,35, 36,42,22, 48,36,44,4 1,41 \rangle$.

Кортеж весов вершин графа G_3 : $f_v(G_3) = \langle 117,210,16 3,106,106, 210,163,11 7 \rangle$.

Базовый инвариант графа:

$$F_e(G_3) \& F_v(G_3) = (22,35,35, 36,36,41,4 1,41,41,42, 42,44,44, 48,48) \& (106,106,1 17,117,163, 163,210,2 10)$$

Количество уровней в спектре реберных разрезов графа $G_4 = 6$. Кортежи весов ребер:

- уровня 1: $\xi(w(r_1)) = \langle 6,6,5,7,5, 5,6,8,6,6, 6,6,7,5,5 \rangle$;
- уровня 2: $\xi(w(r_2)) = \langle 8,7,6,9,9, 6,6,6,8,8, 6,7,9,9,8 \rangle$;
- уровня 3: $\xi(w(r_3)) = \langle 5,7,9,6,9, 9,11,12,7, 7,11,7,6,9, 5 \rangle$;
- уровня 4: $\xi(w(r_4)) = \langle 8,6,8,10,6, 8,8,0,8,8, 8,6,10,6, 8 \rangle$;
- уровня 5: $\xi(w(r_5)) = \langle 6,10,6,12, 10,6,6,0,6, 6,6,10,12, 10,6 \rangle$;
- уровня 6: $\xi(w(r_6)) = \langle 12,12,12,0, 12,12,12, 0,12,12,12, 12,0,12,1 2 \rangle$.

Суммарный кортеж весов ребер графа G_4 : $f_e(G_4) = \langle 44,48,46,4 4,51,46,49, 26,47,47, 49,48,44,5 1,44 \rangle$.

Кортеж весов вершин графа G_4 : $f_v(G_4) = \langle 138,185,21 4,147,214, 147,138,18 5 \rangle$.

Базовый инвариант графа:

$$F_e(G_4) \& F_v(G_4) = (26,44,44, 44,44,46,4 6,47,47,48, 48,49,49, 51,51) \& (138,138,1 47,147,185, 185,214,2 14)$$

Изоспектральные графы G_3 и G_4 не изоморфны, так как их базовые инварианты не совпадают.

I. ВЫДЕЛЕНИЕ ИЗОМОРФНОГО ПОДГРАФА В НЕОРИЕНТИРОВАННОМ ГРАФЕ

Рассмотрим задачу выделения изоморфного подграфа. Пусть заданы два графа $G = (V, E)$ и $H = (X, U)$ причем $|V| \geq |X|$. Существует ли подграф $V_0 \subseteq V, E_0 \subseteq E \cap (V_0 \times V_0)$, такой, что $G_0 \cong H$. Т.е. существует ли отображение $(v_1, v_2) \in E_0 \Leftrightarrow (f(v_1), f(v_2)) \in U$?

Полный инвариант графа включает в себя часть под названием базовый инвариант, не зависящую от нумерации вершин и ребер графа. В связи с этим появляется возможность разбить задачу поиска подграфа, изоморфного заданному графу H , на части (метод диакоптики) [5] в следующей последовательности:

- 1) построение полного инварианта для эталонного графа H ;
- 2) построение упорядоченного вектора локальных степеней графа H для сравнения с упорядоченным вектором локальных степеней подграфа G_0 ;

3) используя алгоритмы сочетания элементов, отобрать подграфы графа G с эквивалентными упорядоченными векторами степеней с количеством вершин и ребер равными эталонному графу H;

4) для каждого отобранного подграфа G₀ графа G, вычислить базовый инвариант и в случае совпадения с базовым инвариантом графа H объявить подграф G₀ изоморфным эталонному графу H.

Например, для графов H и G, представленных на рис. 5, имеем: $n_h = 7, m_h = 13, n_g = 9, m_g = 19$, векторы локальных степеней – $s(H) = (3,3,3,3,4,5,5)$, $s(G) = (3,3,3,4,4,5,5,5,6)$. Очевидно, что для сравнения графов нужно добиться, чтобы количества вершин в подграфе G₀ и в эталоне H были равными. Этого равенства можно достичь, исключив из графа G ровно $k = n_g - n_h$ вершин, проведя $n_g!/(k!(n_g-k)!)$ операций сочетаний вершин. Очевидно, что в процессе перечисления вершин для удаления должны выделяться те подграфы G₀, у которых вектор локальных степеней больше вектора локальных степеней эталона.

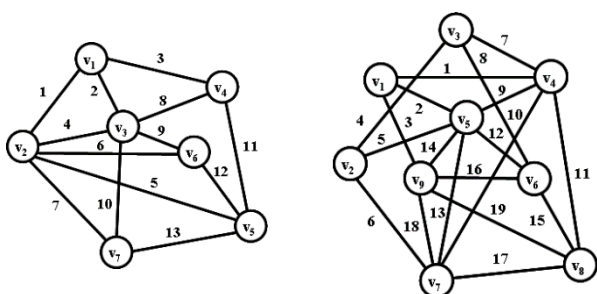


Рис. 4. Графы H и G

Определение 1. При сравнении двух векторов локальных степеней с равной мощностью, большим вектором локальных степеней считается тот, для всех элементов которого выполняется условие:

$$|s(g_1)| \geq |s(h_1)|, |s(g_2)| \geq |s(h_2)|, \dots, |s(g_n)| \geq |s(h_n)| \quad (1)$$

Если существует хотя бы одно не выполненное условие – то вектор меньше.

Сгенерируем всевозможные сочетания для удаления вершин $C_{n_g}^k$ [4]. Для каждого сочетания составим таблицу изменения векторов локальных степеней вершин, располагая их в порядке невозрастания.

Таблица 1

Выделение подграфов

удаляемые вершины	вектор локальных степеней	кол. ребер
граф H	(3,3,3,3,4,5,5)	13
$s(G \setminus \{v_1, v_2\})$	(2,4,4,4,4,4,4)	13
$s(G \setminus \{v_1, v_3\})$	(3,3,3,4,4,5,5)	13
$s(G \setminus \{v_1, v_4\})$	(2,3,3,4,4,4,4)	12
$s(G \setminus \{v_1, v_5\})$	(2,3,3,3,3,4,4)	11
$s(G \setminus \{v_1, v_6\})$	(2,3,3,3,4,4,5)	12

$s(G \setminus \{v_1, v_7\})$	(2,3,3,3,3,4,4)	12
$s(G \setminus \{v_1, v_8\})$	(3,3,3,3,3,4,5)	12
$s(G \setminus \{v_1, v_9\})$	(3,3,3,3,4,4,4)	12
$s(G \setminus \{v_2, v_3\})$	(3,3,4,4,4,5,5)	14
$s(G \setminus \{v_2, v_4\})$	(2,2,3,3,4,4,5)	11
$s(G \setminus \{v_2, v_5\})$	(2,2,3,3,4,4,4)	11
$s(G \setminus \{v_2, v_6\})$	(1,3,3,4,4,4,5)	12
$s(G \setminus \{v_2, v_7\})$	(2,3,3,4,4,4,4)	12
$s(G \setminus \{v_2, v_8\})$	(2,3,3,3,4,4,5)	12
$s(G \setminus \{v_2, v_9\})$	(2,2,2,3,3,4,5)	11
$s(G \setminus \{v_3, v_4\})$	(2,2,3,3,4,5,5)	12
$s(G \setminus \{v_3, v_5\})$	(1,2,2,3,4,4,4)	10
$s(G \setminus \{v_3, v_6\})$	(2,3,3,4,4,5,5)	13
$s(G \setminus \{v_3, v_7\})$	(1,3,3,3,3,4,5)	12
$s(G \setminus \{v_3, v_8\})$	(2,2,3,3,4,4,6)	12
$s(G \setminus \{v_3, v_9\})$	(2,2,2,3,4,4,5)	12
$s(G \setminus \{v_4, v_5\})$	(2,2,3,3,3,3,5)	12
$s(G \setminus \{v_4, v_6\})$	(1,2,3,3,4,4,4)	12
$s(G \setminus \{v_4, v_7\})$	(1,2,2,2,3,4,4)	10
$s(G \setminus \{v_4, v_8\})$	(2,2,3,3,3,4,5)	11
$s(G \setminus \{v_4, v_9\})$	(1,2,2,3,3,3,5)	10
$s(G \setminus \{v_5, v_6\})$	(2,2,2,3,3,4,4)	10
$s(G \setminus \{v_5, v_7\})$	(1,2,3,3,3,3,3)	9
$s(G \setminus \{v_5, v_8\})$	(2,2,2,3,3,3,4)	10
$s(G \setminus \{v_5, v_9\})$	(1,2,2,3,3,3,4)	9
$s(G \setminus \{v_6, v_7\})$	(2,2,2,3,3,4,4)	10
$s(G \setminus \{v_6, v_8\})$	(2,3,3,3,4,4,5)	12
$s(G \setminus \{v_6, v_9\})$	(2,2,2,3,4,5,5)	12
$s(G \setminus \{v_7, v_8\})$	(2,3,3,3,3,3,5)	11
$s(G \setminus \{v_7, v_9\})$	(2,2,2,3,3,4,4)	10
$s(G \setminus \{v_8, v_9\})$	(2,3,3,3,3,4,5)	12

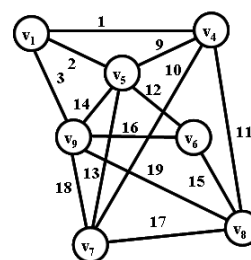


Рис. 5. Подграф графа G после удаления вершин {v₂, v₃}

В нашем примере:

$$s(G \setminus \{v_1, v_2\}) = (2,4,4,4,4,4,4) < s(H) = (3,3,3,3,4,4,5);$$

$$s(G \setminus \{v_1, v_3\}) = (2,3,3,4,4,4,5) < s(H) = (3,3,3,3,4,4,5);$$

$$\dots \dots \dots$$

$$s(G \setminus \{v_2, v_3\}) = (3,3,4,4,4,5,5) > s(H) = (3,3,3,3,4,4,5);$$

$$\dots \dots \dots$$

$$s(G \setminus \{v_7, v_9\}) = (2,2,2,3,3,4,4) < s(H) = (3,3,3,3,4,4,5);$$

$$s(G \setminus \{v_8, v_9\}) = (2,3,3,3,3,4,5) < s(H) = (3,3,3,3,4,4,5).$$

Из всех удаленных пар вершин выбираем $\{v_2, v_3\}$, так как вектор локальных степеней после удаления этих вершин из графа больше вектора локальных степеней эталона. Количество ребер в подграфе равно 14, а в эталоне 13. Следовательно, нужно удалить одно ребро.

Таблица 2

Удаление ребер в подграфе

удаляемое ребро	вектор локальных степеней
граф Н	(3,3,3,3,4,5,5)
$s(G \setminus \{v_2, v_3, e_1\})$	(2,3,3,4,4,5,5)
$s(G \setminus \{v_2, v_3, e_2\})$	(2,3,4,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_3\})$	(2,3,4,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_9\})$	(3,3,3,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_{10}\})$	(3,3,3,3,4,5,5)
$s(G \setminus \{v_2, v_3, e_{11}\})$	(3,3,3,3,4,5,5)
$s(G \setminus \{v_2, v_3, e_{12}\})$	(2,3,4,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_{13}\})$	(3,3,3,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_{14}\})$	(3,3,4,4,4,4,4)
$s(G \setminus \{v_2, v_3, e_{15}\})$	(2,3,3,4,4,5,5)
$s(G \setminus \{v_2, v_3, e_{16}\})$	(2,3,4,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_{17}\})$	(3,3,3,3,4,5,5)
$s(G \setminus \{v_2, v_3, e_{18}\})$	(3,3,3,4,4,4,5)
$s(G \setminus \{v_2, v_3, e_{19}\})$	(3,3,3,4,4,4,5)

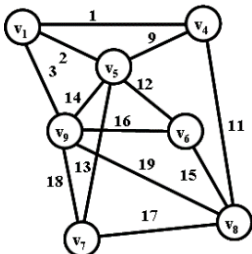


Рис. 6. Подграф G_1 без ребра e_{10}

В результате мы получили 3 варианта решения при удалении ребер e_{10}, e_{11} и e_{17} (рис. 6 – 8). Проверим подграфы на изоморфизм с эталоном [9]. Базовый инвариант эталонного графа Н:

$$F_e(H) \& F_v(H) = (31, 32, 32, 35, 35, 35, 36, 38, 38, 38, 39, 39) \& (98, 102, 115, 115, 144, 176, 182) = (31, 2 \times 32, 3 \times 35, 36, 4 \times 38, 2 \times 39) \& (98, 102, 2 \times 115, 144, 176, 182)$$

Базовый инвариант подграфа G_1 :

$$F_e(G_1) \& F_v(G_1) = (31, 2 \times 32, 3 \times 35, 36, 4 \times 38, 2 \times 39) \& (98, 102, 2 \times 115, 144, 176, 182)$$

Базовый инвариант подграфа G_2 :

$$F_e(G_2) \& F_v(G_2) = (8 \times 84, 4 \times 92, 96) \& (4 \times 260, 352, 2 \times 440)$$

Базовый инвариант подграфа G_3 :

$$F_e(G_3) \& F_v(G_3) = (31, 2 \times 32, 3 \times 35, 36, 4 \times 38, 2 \times 39) \& (98, 102, 2 \times 115, 144, 176, 182)$$

В результате получаем 2 варианта решения задачи: подграфы G_1 и G_3 . Следует заметить, что алгоритм перебора всех сочетаний имеет временную сложность $O(n_g! / k!(n_g - k)!)$. Поэтому для сокращения вычислений нужно стремиться, чтобы величина k (разность количества вершин) была как можно меньшей или как можно большей. Полный инвариант графа Н: количество вершин графа = 7; количество ребер графа = 13; кортеж весов ребер уровня 1: $\langle 6, 6, 4, 8, 7, 6, 6, 6, 6, 6, 5, 5, 5 \rangle$; кортеж весов ребер уровня 2: $\langle 7, 6, 7, 5, 7, 6, 6, 5, 7, 7, 3, 7, 7 \rangle$; кортеж весов ребер уровня 3: $\langle 9, 6, 7, 3, 9, 7, 7, 5, 8, 8, 7, 8, 8 \rangle$; кортеж весов ребер уровня 4: $\langle 5, 6, 9, 7, 5, 8, 8, 3, 7, 9, 7, 7 \rangle$; кортеж весов ребер уровня 5: $\langle 3, 4, 5, 7, 3, 6, 6, 5, 5, 5, 5, 5 \rangle$; кортеж весов ребер уровня 6: $\langle 5, 4, 3, 5, 5, 5, 7, 6, 6, 3, 6, 6 \rangle$; кортеж весов ребер: $\langle 35, 32, 35, 35, 36, 38, 38, 38, 31, 39, 39, 32, 38, 38 \rangle$; кортеж весов вершин: $\langle 102, 182, 176, 98, 144, 115, 115 \rangle$; вектор весов ребер: $(31, 32, 32, 35, 35, 35, 36, 38, 38, 38, 39, 39)$; вектор весов вершин: $(98, 102, 115, 115, 144, 176, 182)$.

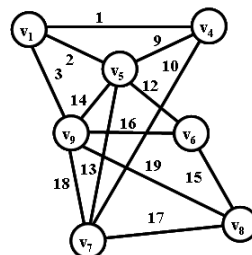


Рис. 7. Подграф G_2 без ребра e_{11}

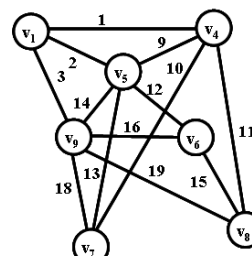


Рис. 8. Подграф G_3 без ребра e_{17}

Полный инвариант подграфа G_1 : количество вершин подграфа = 7; количество ребер подграфа = 13; кортеж весов ребер уровня 1: $\langle 6, 6, 4, 8, 7, 6, 6, 6, 6, 6, 5, 5, 5 \rangle$; кортеж весов ребер уровня 2: $\langle 7, 6, 7, 5, 7, 6, 6, 5, 7, 7, 3, 7, 7 \rangle$; кортеж весов ребер уровня 3: $\langle 9, 6, 7, 3, 9, 7, 7, 5, 8, 8, 7, 8, 8 \rangle$; кортеж весов ребер уровня 4: $\langle 5, 6, 9, 7, 5, 8, 8, 3, 7, 9, 7, 7 \rangle$; кортеж весов ребер уровня 5: $\langle 3, 4, 5, 7, 3, 6, 6, 5, 5, 5, 5, 5 \rangle$; кортеж весов ребер уровня 6: $\langle 5, 4, 3, 5, 5, 5, 7, 6, 6, 3, 6, 6 \rangle$; кортеж весов ребер: $\langle 35, 32, 35, 35, 36, 38, 38, 38, 31, 39, 39, 32, 38, 38 \rangle$; кортеж весов вершин: $\langle 102, 182, 176, 98, 144, 115, 115 \rangle$; вектор весов

ребер: (31,32,32,35,35,35,36,38,38,38,38,39,39); вектор весов вершин: (98,102,115,115,144,176,182).

Полный инвариант подграфа G_3 : количество вершин подграфа = 7; количество ребер подграфа = 13; кортеж весов ребер уровня 1: <6,6,5,8,6,6,6,7,6,6,5,5,4>; кортеж весов ребер уровня 2: <7,6,7,5,5,6,7,7,6,3,7,7>; кортеж весов ребер уровня 3: <8,7,8,3,5,6,8,9,9,7,7,8,7>; кортеж весов ребер уровня 4: <7,8,7,7,3,6,7,5,5,8,9,7,9>; кортеж весов ребер уровня 5: <5,6,5,7,5,4,5,3,3,6,5,5,5>; кортеж весов ребер уровня 6: <6,5,6,5,7,4,6,5,5,5,3,6,3>; кортеж весов вершин: <115,176,182,144,98,102,115>; вектор весов ребер: (31,32,32,35,35,35,36,38,38,38,38,39,39); вектор весов вершин: (98,102,115,115,144,176,182).

II. ВЫДЕЛЕНИЕ ИЗОМОРФНОГО ПОДГРАФА В СЛУЧАЕ СМЕШАННОГО ГРАФА

Решение задачи выделения изоморфного подграфа в случае смешанного графа имеет много особенностей. Рассмотрим пример выделения изоморфного подграфа в случае графа с ориентированными ребрами.

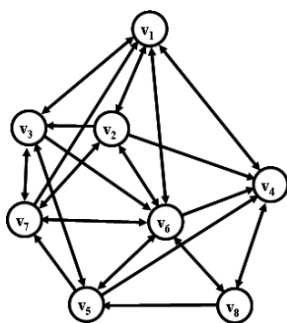
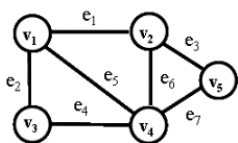
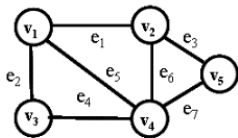


Рис. 9. Граф G

Постановка задачи: существует ли подграф в графе G (рис. 9) изоморфный эталонному графу H (рис. 10)?



а) неориентированный граф;



б) ориентированный граф.

Рис. 10. Эталонный граф H

Определим полный инвариант графа H. Рассмотрим центральные разрезы ориентированного графа H [6,9].

$s_1 = \{e_1, e_2\}$; $s_2 = \{e_1, e_3\}$; $s_3 = \{e_4\}$; $s_4 = \{e_4, e_5, e_6, e_7\}$; $s_5 = \{e_3\}$.

$$\sum = s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5 = \{e_2, e_5, e_6, e_7\}.$$

Формируем множество базовых реберных разрезов:

$$w_1(e_1) = s_1 \oplus s_2 = \{e_2, e_3\};$$

$$w_1(e_2) = s_1 \oplus s_5 = \{e_1, e_2, e_4\};$$

$$w_1(e_3) = s_2 \oplus s_5 = \{e_1\};$$

$$w_1(e_4) = s_3 \oplus s_4 = \{e_5, e_6, e_7\};$$

$$w_1(e_5) = s_1 \oplus s_4 = \{e_1, e_2, e_4, e_5, e_6, e_7\};$$

$$w_1(e_6) = s_2 \oplus s_4 = \{e_1, e_3, e_4, e_5, e_6, e_7\};$$

$$w_1(e_7) = s_4 \oplus s_5 = \{e_3, e_4, e_5, e_6, e_7\}.$$

Реберные разрезы 2-го уровня:

$$w_2(e_1) = \gamma \{e_2, e_3\} = \{e_4, e_5, e_6, e_7\};$$

$$w_2(e_2) = \gamma \{e_1, e_2, e_4\} = \{e_1, e_2, e_3, e_4\};$$

$$w_2(e_3) = \gamma \{e_1\} = \{e_3, e_5, e_6, e_7\};$$

$$w_2(e_4) = \gamma \{e_5, e_6, e_7\} = \{e_4\};$$

$$w_2(e_5) = \gamma \{e_1, e_2, e_4, e_5, e_6, e_7\} = \{e_1, e_3, e_5, e_6, e_7\};$$

$$w_2(e_6) = \gamma \{e_1, e_3, e_4, e_5, e_6, e_7\} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\};$$

$$w_2(e_7) = \gamma \{e_3, e_4, e_5, e_6, e_7\} = \{e_1, e_4, e_5, e_6, e_7\}.$$

Здесь $\{e_2, e_5, e_6, e_7\}$ – сумма всех ориентированных ребер.

Реберные разрезы 3-го уровня:

$$w_3(e_1) = \gamma \{e_4, e_5, e_6, e_7\} = \{e_4, e_5, e_6, e_7\} = \emptyset;$$

$$w_3(e_2) = \gamma \{e_1, e_2, e_4\} = \{e_1, e_2, e_3, e_4\} = \emptyset;$$

$$w_3(e_3) = \gamma \{e_1\} = \{e_3, e_5, e_6, e_7\} = \emptyset;$$

$$w_3(e_4) = \gamma \{e_5, e_6, e_7\} = \{e_4\} = \emptyset;$$

$$w_3(e_5) = \gamma \{e_1, e_2, e_4, e_5, e_6, e_7\} = \{e_1, e_3, e_5, e_6, e_7\} = \emptyset;$$

$$w_3(e_6) = \gamma \{e_1, e_3, e_4, e_5, e_6, e_7\} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} = \emptyset;$$

$$w_3(e_7) = \gamma \{e_3, e_4, e_5, e_6, e_7\} = \{e_1, e_4, e_5, e_6, e_7\} = \emptyset.$$

Кортеж весов ребер 1-го уровня <4,3,3,4,4,4,4>. Кортеж весов ребер 2-го уровня <4,2,4,5,5,5,5>. Суммарный кортеж весов <8,5,7,9,9,9,9>. Кортеж весов вершин <22,24,14,36,16>. Вектор валентности эталонного неориентированного подграфа H – (2,2,3,3,4). Вектор валентности ориентированного эталонного графа H – (1,1,2,2,4). Количество дуг в эталонном графе = 4. Количество неориентированных ребер = 3

Будем последовательно удалять заданное количество вершин графа G. Формируем вектор валентности (локальных степеней) как для неориентированного подграфа H, так и для ориентированного подграфа H. Будем рассматривать только случаи, когда вектора валентности подграфов не меньше эталонного графа и при этом нас будут интересовать только случаи совпадения удаленных вершин для неориентированных подграфов и ориентированных подграфов (табл. 3).

Число операций для выделения неориентированных подграфов = 31. Число операций для подграфов с ориентированными ребрами = 24.

Будем рассматривать только случаи совпадения удаленных вершин, когда вершины выделенного неориентированного подграфа совпадают с вершинами выделенного ориентированного подграфа, так как только в

этом случае удастся вычислить количество смешанных ребер для удаления.

Таблица 3

Множество сочетаний и выделение подграфа

№	Удаленные вершины	Вектор валентности неориентированного подграфа	Вектор валентности ориентированного подграфа	Количество удаляемых ребер
1	v ₁ ,v ₂ ,v ₃	(2,3,3,4,4)		
2	v ₁ ,v ₂ ,v ₄	(2,3,3,4,4)	(1,2,3,3,4)	1
3	v ₁ ,v ₂ ,v ₇	(2,3,3,4,4)		
4	v ₁ ,v ₂ ,v ₈	(2,3,3,4,4)		
5	v ₁ ,v ₃ ,v ₄	(2,2,3,3,4)	(1,2,2,3,4)	0
6	v ₁ ,v ₃ ,v ₅	(2,2,3,3,4)		
7	v ₁ ,v ₃ ,v ₇	(2,3,3,4,4)	(1,2,2,3,4)	1
8	v ₁ ,v ₃ ,v ₈	(3,3,3,3,4)		
9	v ₁ ,v ₄ ,v ₅		(1,2,2,3,4)	
1	v ₁ ,v ₄ ,v ₇	(2,2,3,3,4)	(1,1,2,3,4)	0
1	v ₁ ,v ₄ ,v ₈	(3,3,4,4,4)	(2,2,3,4,4)	2
1	v ₁ ,v ₅ ,v ₇	(2,2,3,3,4)		
1	v ₁ ,v ₅ ,v ₈	(2,3,3,4,4)		
1	v ₁ ,v ₇ ,v ₈	(3,3,3,3,4)		
1	v ₂ ,v ₃ ,v ₄	(2,2,3,3,4)	(1,2,2,2,4)	0
1	v ₂ ,v ₃ ,v ₅	(2,2,3,3,4)		
1	v ₂ ,v ₃ ,v ₇	(2,3,3,4,4)	(2,2,2,3,4)	1
1	v ₂ ,v ₃ ,v ₈	(3,3,3,3,4)		
1	v ₂ ,v ₄ ,v ₅		(1,2,2,3,4)	
2	v ₂ ,v ₄ ,v ₇	(2,2,3,3,4)	(1,2,2,3,4)	0
2	v ₂ ,v ₄ ,v ₈	(3,3,4,4,4)	(2,3,3,3,4)	2
2	v ₂ ,v ₅ ,v ₇	(2,2,3,3,4)		
2	v ₂ ,v ₅ ,v ₈	(2,3,3,4,4)	(2,2,2,3,4)	1
2	v ₂ ,v ₇ ,v ₈	(3,3,3,3,4)		
2	v ₃ ,v ₄ ,v ₅		(1,2,3,3,4)	
2	v ₃ ,v ₄ ,v ₇		(1,2,2,2,4)	
2	v ₃ ,v ₄ ,v ₈	(2,3,3,4,4)	(1,3,3,3,4)	1
2	v ₃ ,v ₅ ,v ₇	(2,3,3,4,4)	(2,2,3,3,4)	1
2	v ₃ ,v ₅ ,v ₈	(3,3,4,4,4)	(2,3,3,3,4)	2
3	v ₃ ,v ₇ ,v ₈	(2,3,3,4,4)	(1,2,3,3,4)	1
3	v ₄ ,v ₅ ,v ₇		(1,2,2,3,4)	
3	v ₄ ,v ₅ ,v ₈	(4,4,4,4,4)	(3,3,3,4,4)	3
3	v ₄ ,v ₆ ,v ₈	(2,3,3,4,4)	(1,2,3,3,4)	1
3	v ₄ ,v ₇ ,v ₈	(2,3,3,4,4)	(2,2,3,3,4)	1
3	v ₅ ,v ₆ ,v ₈	(2,3,3,4,4)	(2,2,2,3,4)	1
3	v ₅ ,v ₇ ,v ₈	(3,3,4,4,4)	(2,2,3,3,4)	2

Рассмотрим только первые четыре случая удаления ребра. Удаляем вершины v₁,v₂,v₄ (рис. 11). Количество

ребер в подграфе = 8, нужно удалить 1 ребро. Однако количество дуг в графе = 3. Поэтому выделенный подграф невозможно привести к эталонному. Удаляем вершины v₁,v₃,v₄ (рис. 12). Количество ребер в подграфе = 7, нужно удалить 0 ребер. Однако количество дуг в графе = 2. Поэтому выделенный подграф невозможно привести к эталонному. Удаляем вершины v₁,v₃,v₇ (рис. 13). Количество ребер в подграфе = 8, нужно удалить 1 ребро. Количество дуг в подграфе = 4. Выделенный подграф можно привести к эталонному удалив ребро (v₆,v₈). Удаляем вершины v₁,v₄,v₅ (рис. 14). Данный случай исключается из рассмотрения, так как существует неориентированный подграф имеющий ребро с валентностью 1. Выделенный подграф невозможно привести к эталонному. Впредь такие случаи, когда отсутствует выделение неориентированного подграфа, мы рассматривать не будем.

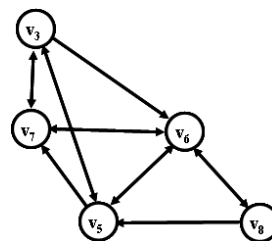


Рис. 11. Подграф с удаленными вершинами v₁,v₂,v₄

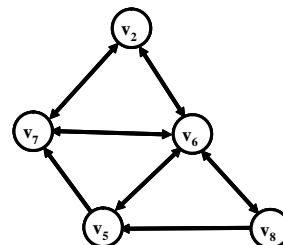


Рис. 12. Подграф с удаленными вершинами v₁,v₃,v₄

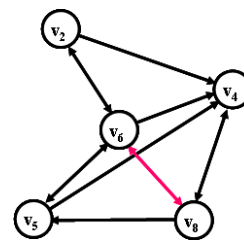


Рис. 13. Подграф с удаленными вершинами v₁,v₃,v₇

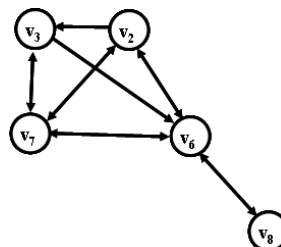


Рис. 14. Подграф с удаленными вершинами v₁,v₄,v₅

Перебрав все возможные варианты удаления вершин и ребер получим единственное решение. Удаление вершин v₃,v₅,v₇ и ребра (v₆,v₈), представленное на рис.

13, приводит к выделению подграфа G_0 графа G изоморфного эталонному графу H .

III. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ МЕТОДА

Рассмотрим вычислительную сложность представленного метода с применением полного инварианта графа, основанного на центральных разрезах графа. Вычислительная сложность алгоритма построения полного инварианта графа равна $o(m^5)$ [9]. Число сочетаний из n по k равно биномиальному коэффициенту [4]:

$$\binom{n}{k} = C_n^k = \frac{n!}{k!(n-k)!} \quad (3)$$

Преобразуем выражение

$$C_n^k = \frac{n(n-1)(n-2)\dots(n-k+1)(n-k)\dots 1}{k!(n-k)!} \quad (4)$$

Сократим числитель и знаменатель на величину $(n-k)!$, получим

$$C_n^k = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!} \text{ при } k \leq n-k \quad (5)$$

$$C_n^k = \frac{n(n-1)(n-2)\dots(k+1)}{(n-k)!} \text{ при } k \geq n-k \quad (6)$$

Данное выражение можно усилить

$$C_n^k = \frac{n(n-1)(n-2)\dots(n-k+1)}{(n-k)!} \approx \frac{n^k}{k!} \text{ при } k \leq n-k \quad (7)$$

$$C_n^k = \frac{n(n-1)(n-2)\dots(k+1)}{(n-k)!} \approx \frac{n^{n-k}}{(n-k)!} \text{ при } k \geq n-k \quad (8)$$

Теперь можно сказать, что вычислительная сложность алгоритма построения сочетаний из n элементов по k равна $o(n^k)$ для $k \leq n-k$ и $o(n^{n-k})$ – для $k \geq n-k$.

Вернемся к задаче выделения изоморфного подграфа. Обозначим количество вершин в графе G через n , количество вершин в графе H – n_H . Обозначим количество ребер в подграфе G_0 через m_0 , количество ребер в графе H – m_H . Тогда вычислительную сложность алгоритма выделения изоморфного подграфа с учетом разделения процессов удаления вершин и удаления ребер можно оценить как:

$$o(n^{n_H} \times m_0^{m_H}) + o(m_H^5). \quad (12)$$

Если произвести дальнейшее огрубление введя переменную $p = \max(n, m_0)$, то вычислительную сложность можно представить в виде:

$$o(p^{n_H + m_H}) + o(m_H^5). \quad (13)$$

Представленный метод имеет полиномиальную вычислительную сложность.

ЛИТЕРАТУРА

- [1] Курейчик В.М., Глушань В.М., Щербаков Л.И. Комбинаторные аппаратные модели и алгоритмы в САПР. М.: Радио и связь, 1990. 216 с.
- [2] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. / М. Гэри., Д. Джонсон. М.: Мир, 1982. 416 с.
- [3] Герман О.В., Дунаев А.А. Задача изоморфизма графов в системе нечеткого распознавания / Труды БГТУ. 2016. № 6. С. 181–184.
- [4] Knuth D.E. The Art of Computer Programming. V. 4a. "Combinatorial Algorithms". P. 1, 3rd ed. Addison Wesley, 1997. 912 p.
- [5] Крон Г. Исследование сложных систем по частям (диакоптика). М.: Наука, 1972. 544 с.
- [6] Курапов С.В., Давидовский М.В., Неласая А.В. Изоморфизм графов. Монография. Запоріжжя: ЗНУ, 2019. 120 с.
- [7] Черноухов С.А. Проверка изоморфности двух графов и поиск изоморфных подграфов: подход, основанный на анализе максимально протяженных неразветвляющихся путей. Препринт, Санкт-Петербург, 2020.
- [8] Cvetkovic D.M., Doob M., Sachs H. Spectra of graphs. Theory and application. Heidelberg/Leipzig, Johann Ambrosius Barth Verlag, 1984. 447 p.
- [9] Kurapov S.V., Davidovsky M.V. Computational methods for determining graph invariants // International Journal of Open Information Technologies. 2021. V. 9. № 2. P. 1–8.
- [10] Bonnici V., Giugno R., Pulvirenti A. et al. A subgraph isomorphism algorithm and its application to biochemical data // BMC Bioinformatics. 2013. V. 14.
- [11] Cordella L., Foggia P., Sansone C., Vento M. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. V. 26. № 10. P. 1367–1372.
- [12] Ullmann J.R. An algorithm for Subgraph Isomorphism // Journal of the Association for Computing Machinery. 1976. V. 23. P. 31–42.
- [13] Ullmann J.R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism // J Exp Algorithmics. 2010. V. 15. P. 1–64.

A Method for Selecting an Isomorphic Subgraph of a Circuit Diagram Graph in Electronic Circuit CAD Systems

S.V. Kurapov¹, M.V. Davidovsky²

¹Zaporizhzhya National University, Zaporizhzhya, Ukraine, lilili5050@rambler.ru

²Zaporizhzhya Institute of Postgraduate Pedagogical Education, Zaporizhzhya, Ukraine, m.davidovsky@gmail.com

Abstract — Studying the properties of objects based on the analysis of their structural characteristics constitute one of the most important and promising directions in the development of applied graph theory. In general, discrete graph models represent the structures of objects of a very different nature. Analysis of structural characteristics of the designed objects becomes notably important in the design of microcircuits, printed circuit boards, flexible printed circuit boards, integrated circuits, as well as LSI, VLSI and other flat constructs design processes. Thus, a challenging task within the scope of modern CAD systems development activities is the problem of recognizing the structures of various designed objects. In order to increase the efficiency of CAD, in practice, there is a need to use graph isomorphism testing – for example, to verify various representations of an electronic circuit when automating electronic circuits design. The article is devoted to applied mathematical methods for determining the isomorphism of subgraphs of circuit diagram graphs in electronic equipment CAD systems. In this work the authors present a method for solving the problem of extracting a subgraph G_0 from graph $G(V,E)$, which is isomorphic to a given reference graph $H(X,U)$. The method is based on the use of a complete invariant of a graph representing its individual structure in numerical form. Preliminary determination of the complete invariant for the reference graph allows reducing the problem to finding a set of subgraphs of the graph G with the number of vertices equal to the cardinality $\text{card}(X)$ of the set of vertices of the graph H . The proposed solution consists in enumerating all sets of $|X|$ -combinations from $|V|$. Such a problem statement allows applying computational methods with polynomial computational complexity to solve the original hard problem.

Keywords — computer-aided design system, electrical schematic diagram graph, graph isomorphism, isomorphic subgraph extraction.

REFERENCES

- [1] Kurejchik V.M., Glushan' V.M., Shcherbakov L.I. Kombinatornye apparatnye modeli i algoritmy v SAPR (Combinatorial hardware models and algorithms in CAD). M.: Radio i svyaz', 1990. 216 s.
- [2] Garey M.R., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979. 338 p.
- [3] German O.V., Dunaev A.A. Zadacha izomorfizma grafov v sisteme nechetkogo raspoznavaniya (Graph isomorphism problem in the fuzzy recognition system) / Trudy BGTU. 2016. № 6. S. 181–184.
- [4] Knuth D.E. The Art of Computer Programming. V. 4a. "Combinatorial Algorithms". P. 1, 3rd ed. Addison Wesley, 1997. 912 p.
- [5] Kron G. Diakoptics; the piecewise solution of large-scale systems. London, Macdonald, 1963. 166 p.
- [6] Kurapov S.V., Davidovskij M.V., Nelasaya A.V. Izomorfizm grafov (Graph isomorphism). Monografiya. Zaporizhzhya: ZNU, 2019. 120 s.
- [7] Chernouhov S.A. Proverka izomorfnosti dvuh grafov i poisk izomorfnyh podgrafov: podhod, osnovannyj na analize maksimal'no protyazhennyh nerazvetvlyayushchihsya putej (Testing the isomorphism of two graphs and finding isomorphic subgraphs: an approach based on the analysis of maximally extended non-branching paths). Preprint, Sant-Peterburg, 2020.
- [8] Cvetkovic D.M., Doob M., Sachs H. Spectra of graphs. Theory and application. Heidelberg/Leipzig, Johann Ambrosius Barth Verlag, 1984. 447 p.
- [9] Kurapov S.V., Davidovsky M.V. Computational methods for determining graph invariants // International Journal of Open Information Technologies. 2021. V. 9. № 2. P. 1–8.
- [10] Bonnici V., Giugno R., Pulvirenti A. et al. A subgraph isomorphism algorithm and its application to biochemical data // BMC Bioinformatics. 2013. V. 14.
- [11] Cordella L., Foggia P., Sansone C., Vento M. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. V. 26. № 10. P. 1367–1372.
- [12] Ullmann J.R. An algorithm for Subgraph Isomorphism // Journal of the Association for Computing Machinery. 1976. V. 23. P. 31–42.
- [13] Ullmann J.R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism // J Exp Algorithmics. 2010. V. 15. P. 1–64.