

МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ СПЕЦИАЛЬНОГО ВИДА В ПЛОСКИХ ГРАФАХ

Диссертация на соискание ученой степени
доктора физико-математических наук
05.13.17 – Теоретические основы информатики

Макаровских Татьяна Анатольевна

Челябинск – 2020

Цель диссертационной работы

- Решение проблемы маршрутизации специального вида в плоских графах

Всевозможные траекторные задачи являются универсальными математическими моделями многих задач оптимизации и управления.

- Эвристические алгоритмы для построения маршрутов (Xie S.Q., Jing Y., Zhige C., Lee M.K., Kwon K.B., Hoefl J., Palekar U.S.);
- Траекторная стабилизация мобильных роботов (ИПУ им. В.А. Трапезникова РАН, г. Москва, Уткин В.А.);
- Управление маршрутизацией и оптимизация (ИПУ им. В.А. Трапезникова РАН, г. Москва, Лазарев А.А.);
- Задачи построения маршрутов в графах (Fleischner H.).
- Задача маршрутизации для вырезания заготовок из листового материала:
 - раскрой-упаковка (Верхотуров В.А., Картак В.М.);
 - задачи маршрутизации режущего инструмента машин листовой резки с ЧПУ (Петунин А.А., Ченцов А.Г. и др.);
 - декомпозиционный метод оптимизации проектирования управляющих программ тепловой резки металла на оборудовании с ЧПУ (Фроловский В.Д.).

1 Раскройные планы без совмещения резов

- Петунин А.А., Ченцов А.Г., Ченцов П.А. К вопросу о маршрутизации перемещений при листовой резке деталей // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. 2017. Т. 10, № 3. С. 25–39.
- Dewil, R., Vansteenwegen, P., Catrysse, D. A review of cutting algorithms for laser cutters // International Journal of Manufacturing Technologies, 2016. Volume 87, P. 1865–1884.

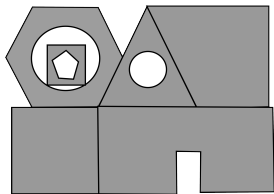
2 Доказано, что эйлеров граф имеет эйлеров цикл, гомеоморфный самонепересекающейся жордановой кривой на плоскости (не учтено условие OE)

- Белый С.Б. О самонепересекающихся и непересекающихся цепях // Математические заметки, 1983. Т. 34. № 4. С. 625–628.
- Manber U., Bent S.W. On Non-intersecting Eulerian Circuits // Discrete Applied Mathematics, Vol. 18, 1987. P. 87–94.

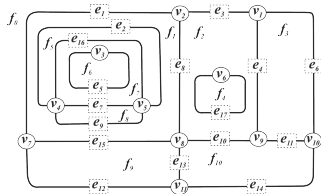
- 3 Построение маршрутов, являющихся A -цепями. Наличие A -цепи решает задачу, но она \mathcal{NP} -трудна (не учтено условие OE)
 - Фляйшнер Г. Эйлеровы графы и смежные вопросы. М.: Мир, 2002. 335 с., ил. (англ. 1990)
- 4 Построение OE -цепей (цепь не является самонепересекающейся)
 - Manber U., Israni S. Pierce Point Minimization and Optimal Torch Path Determination in Flame Cutting // Journal of Manufacturing Systems, Vol. 3, No.1, 1984. P. 81–89.
 - Panioukova T.A., Panyukov A.V. The Algorithm for Tracing of Flat Euler Cycles with Ordered Enclosing // Известия Челябинского научного центра УрО РАН. 2000. № 4(9). P. 18–22.
 - Панюкова Т.А. Обходы с упорядоченным охватыванием в плоских графах // Дискретный анализ и исследование операций. Сер. 2. 2006. Т. 13, № 2. С. 31–43.

- 1 Определить способ представления гомеоморфного образа раскройного плана, позволяющего эффективно решать проблемы маршрутизации;
- 2 Доказать, что в плоских графах существуют маршруты, удовлетворяющие заданному набору локальных и глобальных ограничений специального вида;
- 3 Разработать алгоритмы решения проблемы построения маршрутов специального вида в плоских графах;
- 4 Разработать программное обеспечение для реализации представленных алгоритмов;
- 5 Получить оценки количества полученных маршрутов специального вида.

Абстрагирование раскройного плана до плоского графа



(a) Раскройный план



(b) Гомеоморфный образ

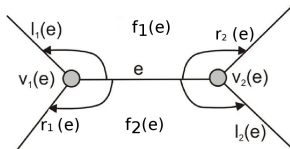
- S – плоскость, модель раскройного листа;
- $G = (V, F, E)$ – плоский граф, модель раскройного плана: V – множество вершин; F – множество граней; E – множество ребер;
- f_0 – внешняя (неограниченная) грань графа G , остальные грани – внутренние;
- $\text{Int}(J)$ для любой части графа $J \subseteq G$ – теоретико-множественное объединение его внутренних граней.
- Если J – пройденная часть маршрута, то $\text{Int}(J)$ – охваченные грани.

Абстрагирование раскройного плана до плоского графа

Предложение

С точностью до гомеоморфизма граф G определяется заданием следующих функций:

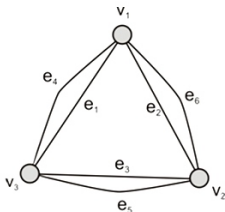
- $v_1(e)$, $v_2(e)$ – вершины, инцидентные ребру e ;
- $f_1(e)$, $f_2(e)$ – грани, находящиеся слева при движении по ребру e из вершины $v_k(e)$ в $v_{3-k}(e)$, $k = 1, 2$;
- $l_1(e)$, $l_2(e)$ – ребра, инцидентные граням $f_k(e)$, $k = 1, 2$ и инцидентные вершинам $v_k(e)$.
- $r_1(e)$, $r_2(e)$ – ребра, инцидентные граням $f_{3-k}(e)$, $k = 1, 2$ и инцидентные вершинам $v_k(e)$.



Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

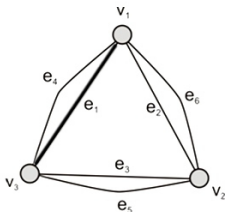
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

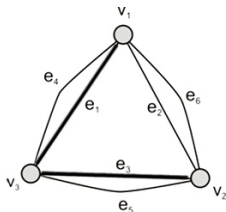
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

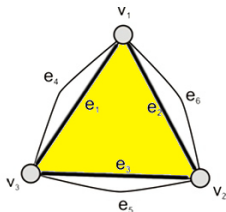
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

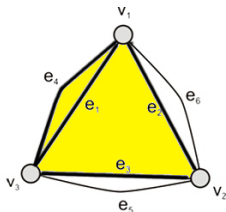
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

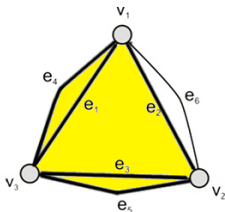
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

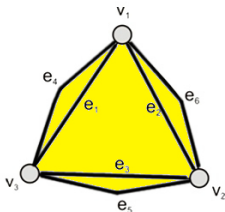
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

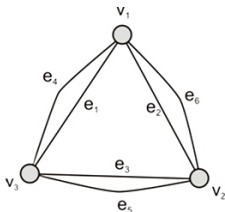
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

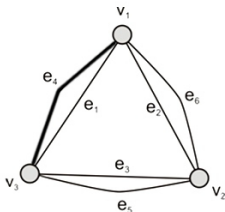
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

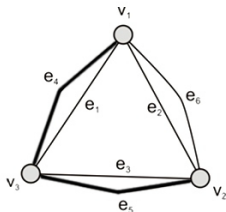
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_2 e_5 v_3 e_6 v_1$ удовлетворяет условию.

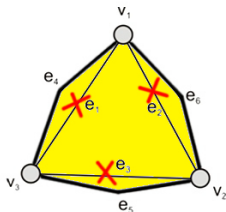
Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1 e_1 v_2 e_3 v_3 e_2 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Цикл с упорядоченным охватыванием (OE -цикл)

Определение

Цикл $C = v_1 e_1 v_2 e_2 \dots v_k$ в эйлеровом графе $G(V, E)$ называется **циклом с упорядоченным охватыванием**, если для любой его начальной части $C_i = v_1 e_1 v_2 e_2 \dots e_i$, $i \leq |E|$ выполнено условие $\text{Int}(C_i) \cap E = \emptyset$.



Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1$ удовлетворяет условию.

Цикл $v_1 e_4 v_2 e_5 v_3 e_6 v_1$ – не удовлетворяет, т.к.

$$\text{Int}(v_1 e_4 v_2 e_5 v_3 e_6 v_1) \supset \{e_1, e_2, e_3\}$$

Теорема 1

Пусть G – плоский эйлеров граф. Для любой вершины $v \in V(G)$, инцидентной границе внешней (бесконечной) грани графа G , существует эйлеров ОЕ-цикл $C = ve_1v_1e_2v_2 \dots v_{|E|-1}e_{|E|}v$.

1

¹Панюкова, Т.А. Обходы с упорядоченным охватыванием в плоских графах / Т.А. Панюкова // Дискретный анализ и исследование операций. Сер. 2. 2006. Т.13, No. 2, с. 31–43.

Покрытие с упорядоченным охватыванием (*OE*-покрытие)

Определение

Упорядоченная последовательность реберно-непересекающихся *OE*-цепей

$$\begin{aligned} C^0 &= v^0 e_1^0 v_1^0 e_2^0 \dots e_{k_0}^0 v_{k_0}^0, & C^1 &= v^1 e_1^1 v_1^1 e_2^1 \dots e_{k_1}^1 v_{k_1}^1, \dots, \\ C^{n-1} &= v^{n-1} e_1^{n-1} v_1^{n-1} e_2^{n-1} \dots e_{k_{n-1}}^{n-1} v_{k_{n-1}}^{n-1}, \end{aligned}$$

покрывающая граф G и такая, что

$$(\forall m : m < n), \quad \left(\bigcup_{l=0}^{m-1} \text{Int}(C^l) \right) \cap \left(\bigcup_{l=m}^{n-1} C^l \right) = \emptyset$$

называется **покрытием с упорядоченным охватыванием** (*OE*-покрытием).

Определение

OE -покрытие минимальной мощности назовем эйлеровым OE -покрытием.

Теорема 2

Пусть G плоский связный граф, $V_{odd}(G)$ – множество вершин нечетной степени графа G , тогда для мощности N эйлера OE -покрытия графа G имеет место неравенство

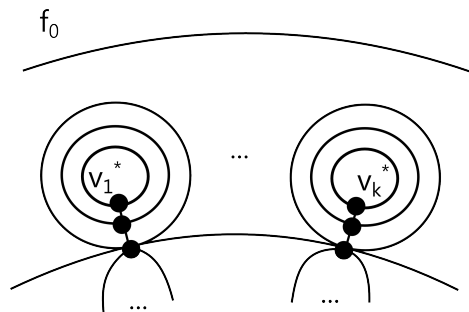
$$k = \frac{|V_{odd}(G)|}{2} \leq N \leq |V_{odd}(G)| = 2k.$$

Верхняя и нижняя границы достижимы.

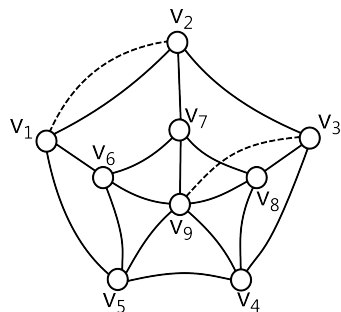
2

²Макаровских Т.А. Оценка мощности OE -покрытия плоского графа // Вестник Уфимского государственного авиационного технического университета. 2017. Т. 21. № 2 (76). С. 112–118.

Примеры графов, для которых достигаются верхняя и нижняя оценки



Верхняя оценка

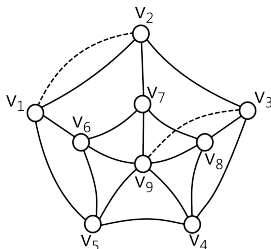


Нижняя оценка

Теорема 3

Пусть $G = (V, E)$ – плоский связный граф на S , не имеющий мостов. Существует множество ребер $H : (H \cap S) \setminus V = \emptyset$ такое, что граф $\hat{G} = (V, E \cup H)$ – эйлеров, и в графе \hat{G} существует эйлеров цикл $C = v_1 e_1 v_2 e_2 \dots e_n v_1$, $n = |E| + |H|$, для любой начальной части которого $C_l = v_1 e_1 v_2 e_2 \dots v_l$, $l \leq |E| + |H|$, выполнено условие $\text{Int}(C_l) \cap G = \emptyset$.

3



³Панюкова, Т.А. Оптимальные эйлеровы покрытия с упорядоченным охватыванием для плоских графов / Т.А. Панюкова // Дискретный анализ и исследование операций. Март–апрель, 2011. – Том 18. – No. 2. – С.64–74.

Теорема 4

Пусть $G = (V, E)$ – плоский связный граф на S , не имеющий мостов, $V_{\text{odd}} \subset V$ – множество вершин нечетной степени. Тогда для любого паросочетания M на множестве вершин V_{odd} в графе $\hat{G} = (V, E \cup M)$ существует эйлеров цикл $C = v_1 e_1 v_2 e_2 \dots e_n v_1$, $n = |E| + |M|$, для любой начальной части которого $C_l = v_1 e_1 v_2 e_2 \dots v_l$, $l \leq |E| + |M|$, выполнено условие $\text{Int}(C_l) \cap G = \emptyset$.

4

⁴Панюкова, Т.А. Оптимальные эйлеровы покрытия с упорядоченным охватыванием для плоских графов / Т.А. Панюкова // Дискретный анализ и исследование операций. Март–апрель, 2011. – Том 18. – No. 2. – С.64–74.

Ранг ребра $e \in E(G)$ – значение функции $\text{rank}(e) : E(G) \rightarrow \mathbb{N}$, определяемую рекурсивно:

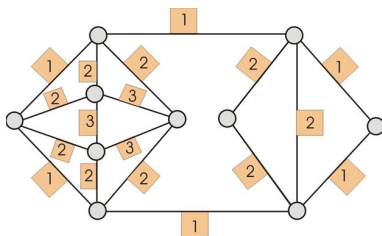
- пусть $E_1 = \{e \in E : e \subset f_0\}$ – множество ребер, ограничивающих внешнюю грань f_0 графа $G(V, E)$, тогда $(\forall e \in E_1) (\text{rank}(e) = 1)$;
- пусть $E_k(G)$ – множество ребер ранга k графа

$$G_k \left(V, E \setminus \left(\bigcup_{l=1}^{k-1} E_l \right) \right),$$

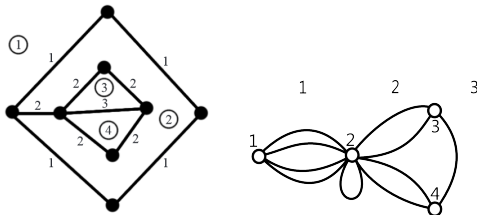
тогда $(\forall e \in E_k) (\text{rank}(e) = k)$.

Способы вычисления $\text{rank}(e)$

- Рекурсивная разметка



- Разметка с помощью поиска в ширину в двойственном графе



Алгоритмы для построения OE-маршрутов

Тип маршрута	Выч. сложность	Комментарий
Эйлеров OE-цикл	$O(V(G) ^2)$	Рекурсивный алгоритм
Эйлеров OE-цикл	$O(E(G) \cdot \log_2 V(G))$	Нерекурсивный алгоритм
Задача китайского почтальона	$O(V(G) ^2)$	
OE-покрытие OE-Cover	$O(E(G) \cdot \log_2 V(G))$	Оптimalен по числу цепей
OE-покрытие (M-Cover)	$O(E(G) \cdot \log_2 V(G))$	Построение OE-покрытия для произвольного паросочетания
Оптимальное OE-покрытие (OptimalCover)	$O(E(G) \cdot \sqrt{ V(G) })$	Оптimalен по числу цепей и длине дополнительных построений
OE-покрытие OE-Cover несвязного графа (MultiComponent)	$O(E(G) \cdot \log_2 V(G))$	При использовании алгоритма OptimalCover – $O(E(G) \cdot \sqrt{ V(G) })$
OE-покрытие OE-Cover несвязного графа (DoubleBridging)	$O(E(G) \cdot \log_2 V(G))$	Если заданы расстояния между компонентами

Алгоритм M-Cover

Вход:

- связный плоский граф G , функции $v_k(e)$, $l_k(e)$, $e \in E(G)$, $k = 1, 2$;
- вершина $v_0 \in V(G)$, инцидентная внешней грани;
- паросочетание M на множестве вершин V_{Odd} нечетной степени;
- булева функция $Idle_M : V_{Odd} \rightarrow \{\text{false}, \text{true}\}$ на множестве V_{Odd} вершин нечетной степени;

Выход: вполне упорядоченное множество C из OE -цепей графа G , представляющее OE покрытие графа G ;

Order (G); // Определить rank() для всех $e \in E(G)$, $v \in V(G)$

$v := v_0$; // Построение

While ($Q(v) \neq \emptyset$)

 FormChain(v , v);

If ($Idle_M(v) \vee (Q(v) = \emptyset)$)

$u \leftarrow M(v)$; // Вершина u является напарником вершины v в паросочетании M

$V_{Odd} \leftarrow V_{Odd} \setminus \{u, v\}$; // Удалить вершины u, v из V_{Odd}

$v \leftarrow u$; // Завершить построение текущей цепи

EndIf

EndWhile

End of algorithm

Входные данные:

- плоский граф G , представленный списком ребер с заданными на них функциями $v_k(e)$, $l_k(e)$, $f_k(e)$, $k = 1, 2$.

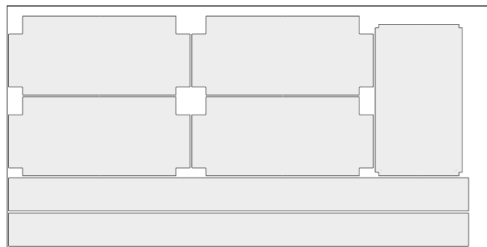
Выходные данные: C_j , $j = 1, \dots, |V_{odd}|/2$, – покрытие графа G OE -цепями.

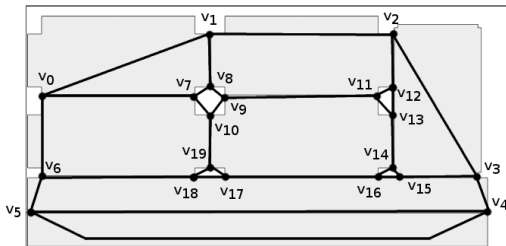
Шаг 1. Найти кратчайшее паросочетание M на множестве V_{odd} .

Шаг 2. Выполнить алгоритм M -Cover для графа G и паросочетания M .

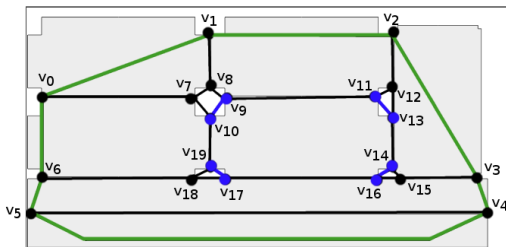
Шаг 3. Останов.

Пример раскройного плана и его оптимального OE -покрытия

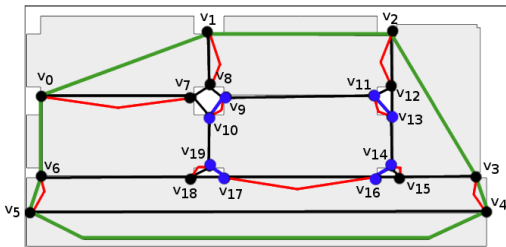




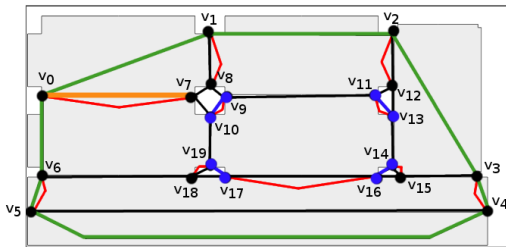
Ранжирование ребер и вершин



Добавление дополнительных ребер

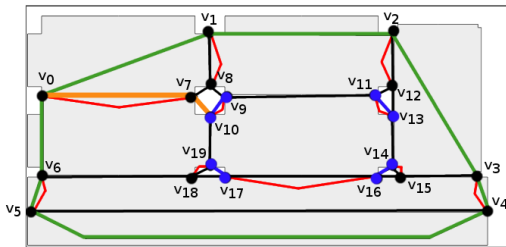


Построение маршрута



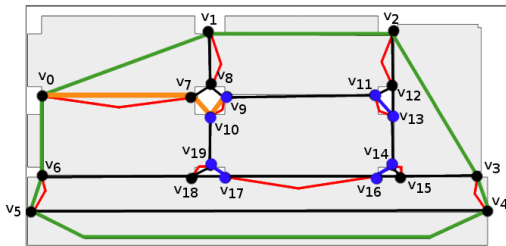
Цепь 1: $v_0 v_7$

Построение маршрута



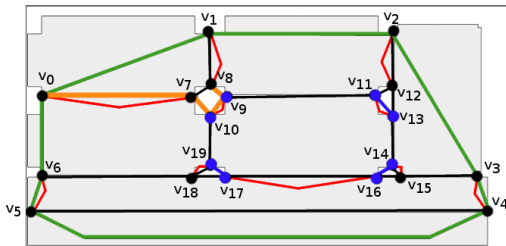
Цепь 1: $v_0 v_7 v_{10}$

Построение маршрута



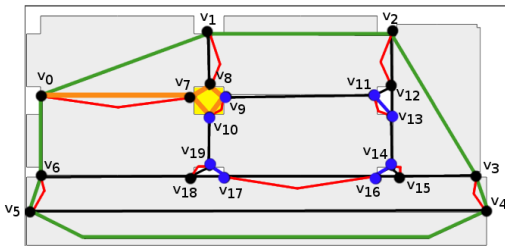
Цепь 1: $v_0 v_7 v_{10} v_9$

Построение маршрута



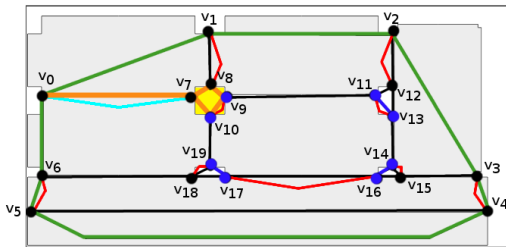
Цепь 1: $v_0 v_7 v_{10} v_9 v_8$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

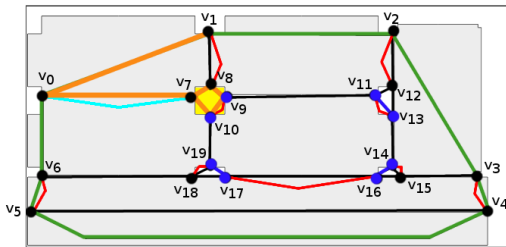
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: v_0

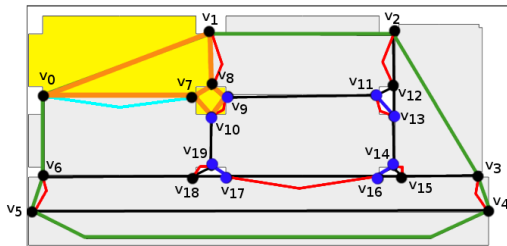
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1$

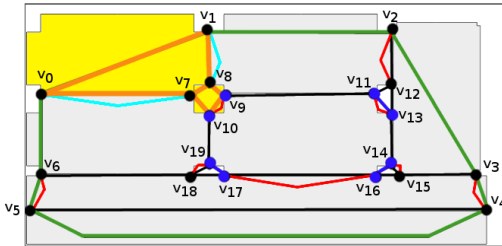
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

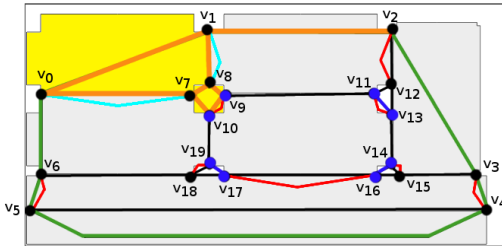
Цепь 2: $v_0 v_1 v_8$

Построение маршрута



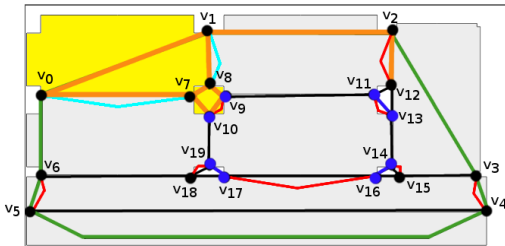
Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$
Цепь 3: v_1

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$
Цепь 3: $v_1 v_2$

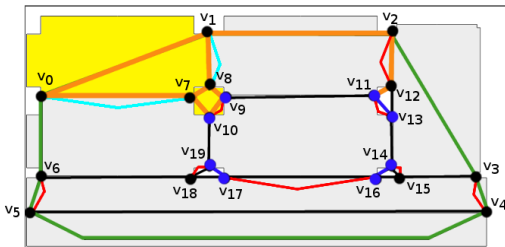
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12}$

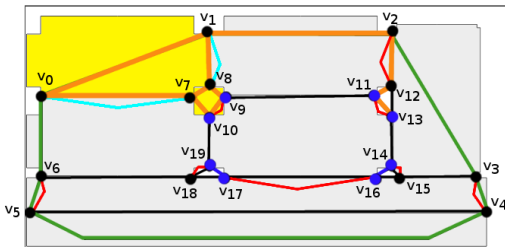
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11}$

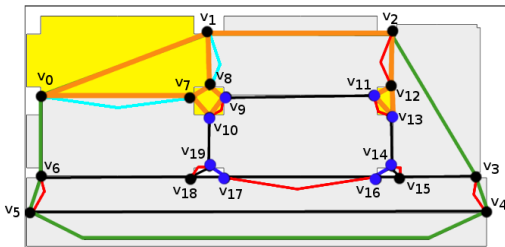
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13}$

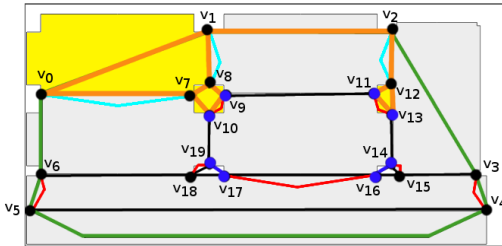
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$

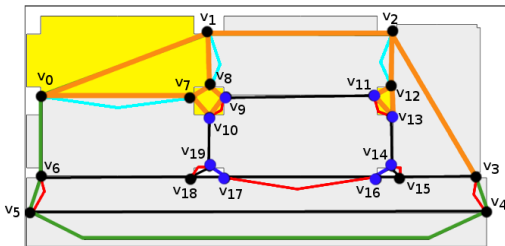
Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$
Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$ Цепь 4: v_2

Построение маршрута



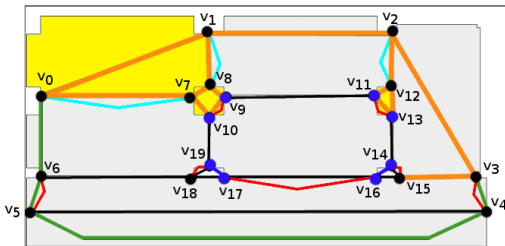
Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3$

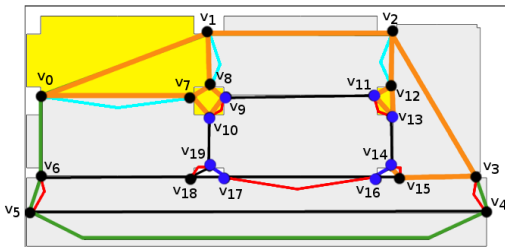
Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$
Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 2: $v_0 v_1 v_8$
Цепь 4: $v_2 v_3 v_{15}$

Построение маршрута



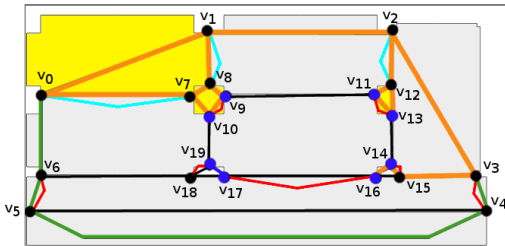
Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14}$

Построение маршрута



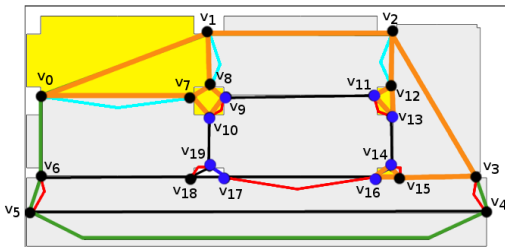
Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16}$

Построение маршрута



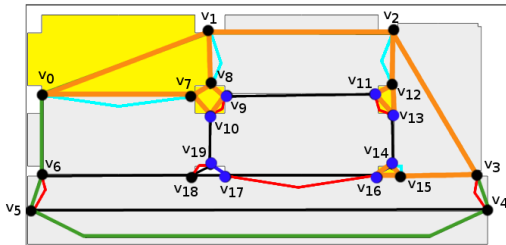
Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

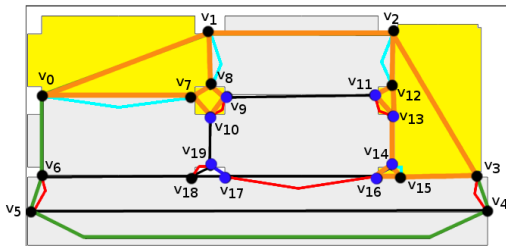
Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: v_{14}

Построение маршрута

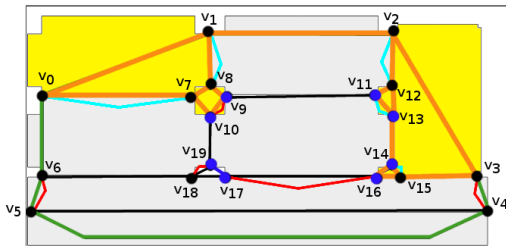


Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$ Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$ Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

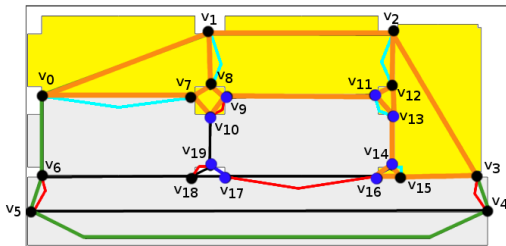
Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: v_{13}

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

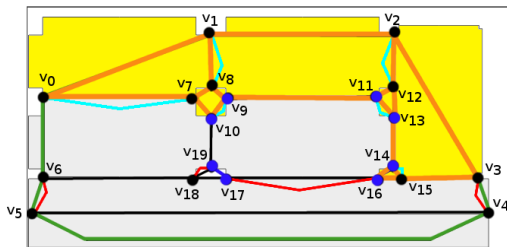
Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

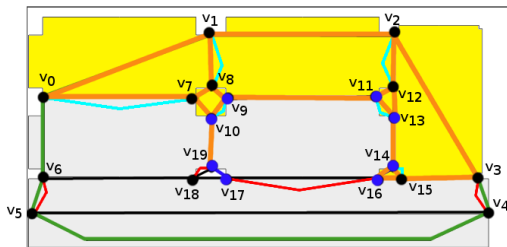
Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: v_{10}

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

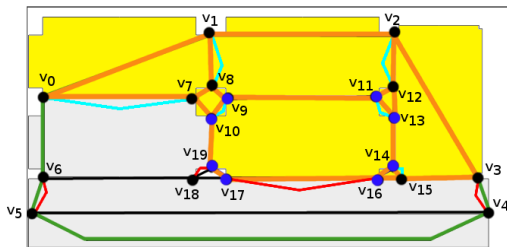
Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

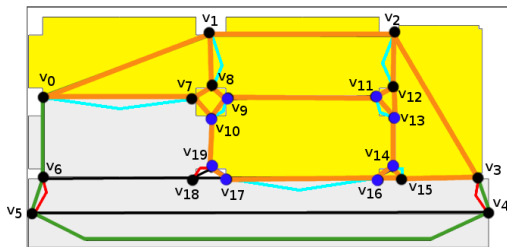
Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

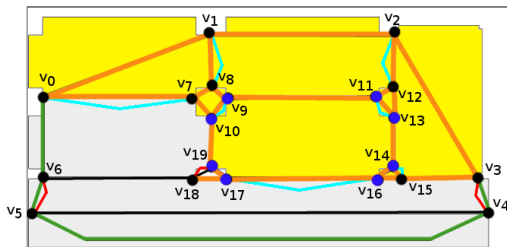
Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: v_{17}

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

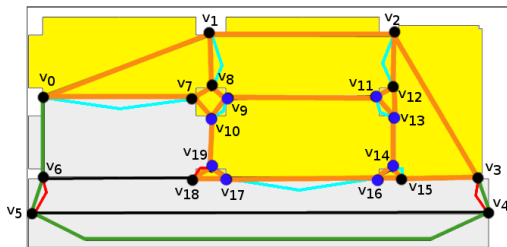
Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

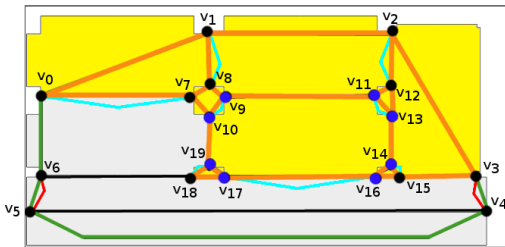
Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

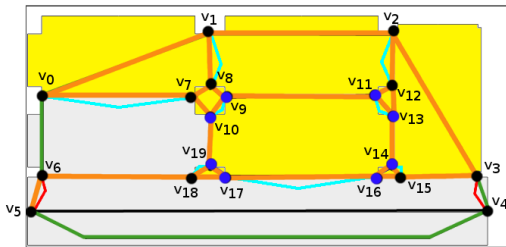
Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Цепь 9: v_{18}

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

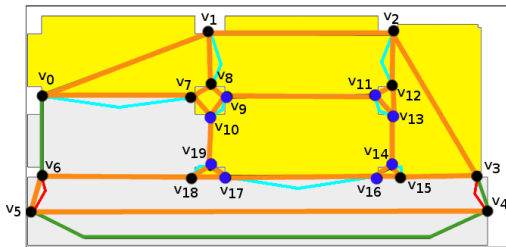
Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Цепь 9: $v_{18} v_6 v_5$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

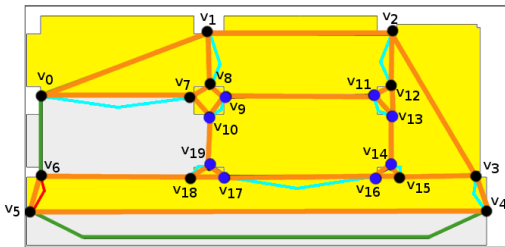
Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Цепь 9: $v_{18} v_6 v_5 v_4$

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

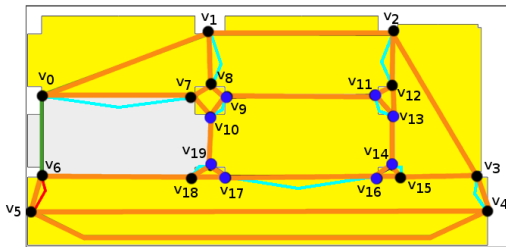
Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Цепь 9: $v_{18} v_6 v_5 v_4 v_3$

Цепь 10: v_4

Построение маршрута



Цепь 1: $v_0 v_7 v_{10} v_9 v_8 v_7$

Цепь 2: $v_0 v_1 v_8$

Цепь 3: $v_1 v_2 v_{12} v_{11} v_{13} v_{12}$

Цепь 4: $v_2 v_3 v_{15} v_{14} v_{16} v_{15}$

Цепь 5: $v_{14} v_{13}$

Цепь 6: $v_{11} v_9$

Цепь 7: $v_{10} v_{19} v_{17} v_{16}$

Цепь 8: $v_{17} v_{18} v_{19}$

Цепь 9: $v_{18} v_6 v_5 v_4 v_3$

Цепь 10: $v_4 v_5$

Определение

Рангом грани $f \in F(G)$ будем называть значение функции $\text{rank} : F(G) \rightarrow \mathbb{Z}^{\geq 0}$:

$$\text{rank}(f) = \begin{cases} 0, & \text{при } f = f_0, \\ \min_{e \in E(f)} \text{rank}(e), & \text{в противном случае,} \end{cases}$$

где $E(f)$ – множество ребер инцидентных грани $f \in F$.

Определение

Ранг компоненты связности – минимальный ранг ребер этой компоненты связности.

Алгоритм MultiComponent

Входные данные:

- несвязный плоский граф G , заданный функциями $v_k(e)$, $f_k(e)$, $l_k(e)$, $e \in E(G)$, $k = 1, 2$;

Выходные данные:

- вполне упорядоченное множество C из OE-цепей графа G , представляющее OE покрытие графа G ;

Промежуточные данные:

- множество $S(G)$ компонент связности графа;

- функции $s(v)$, $s(e) \in S(G)$ определяющие принадлежность вершин и ребер графа компонентам связности $s \in S(G)$;

- множество вершин $v_0(s) \in V(G)$, инцидентных внешней грани для каждой компоненты связности;

- $C(s)$ – OE покрытие компоненты связности $s \in S(G)$; # – символ конца цепи;

Шаг 1. <Поиск> Выявить множество компонент связности $S(G)$.

Шаг 2. <Разметка> Определить ранги всех ребер, вершин, граней и компонент связности графа G . Определить $v_0(s)$ для каждой компоненты связности $s \in S(G)$.

Шаг 3. <Сортировка> Сформировать очередь $Q(S)$ компонент связности $s \in S(G)$ в порядке убывания ранга.

Шаг 4. Пока очередь $Q(S)$ не пуста:

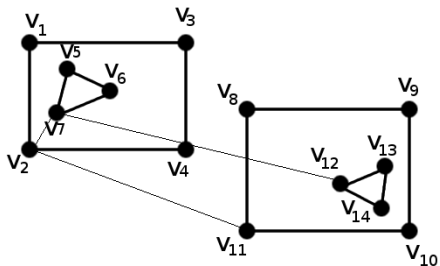
$s \leftarrow Q(S)$; /* переместить из очереди $Q(S)$ первый элемент в переменную s */
перейти на Шаг 4, иначе Останов.

Шаг 5. Найти покрытие $C(s)$: выполнить алгоритм OE-Cover для множества вершин и ребер из s .

$C \leftarrow C(s) \leftarrow \#$. /* завершение текущей цепи и начало следующей */

Конец алгоритма MultiComponent

OE-покрытие несвязного графа



Определение

Грань $f \in F(G)$ будем называть **разделяющей**, если она инцидентна двум и более компонентам связности.

- Пусть граф \tilde{G} получен из графа G добавлением в разделяющих гранях мостов между компонентами связности.
- Полученный граф \tilde{G} будет плоским связным графом и для него может быть построен OE-маршрут $M(\tilde{G})$ с помощью алгоритма OE-Cover.
- Искомый OE-маршрут $M(G)$ может быть получен из маршрута $M(\tilde{G})$ если вершины инцидентные введенным мостам считать окончанием текущей цепи и началом следующей.

OE-покрытие несвязного графа

Алгоритм Bridging

Входные данные: плоский несвязный граф G ;

Выходные данные: плоский связный граф \tilde{G} и множество B добавленных мостов;

Шаг 1. $\tilde{G} := G$; $B = \emptyset$;

Шаг 2. Определить множество C_F всех разделяющих граней.

Шаг 3. ForAll $f \in C_F$

3.1. Найти множество $S(f)$ компонент связности графа G , инцидентных грани f .

3.2. Построить полный абстрактный граф \mathcal{T} , вершинами которого являются компоненты связности $S(f)$, а длины ребер равны расстоянию между соответствующими компонентами.

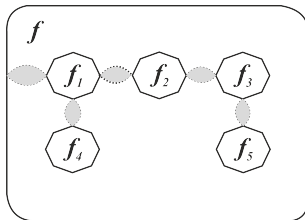
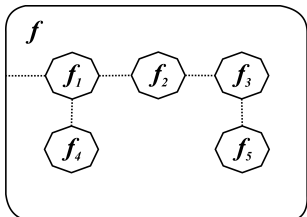
3.3. Найти остовное дерево минимального веса $T(\mathcal{T})$ в \mathcal{T} .

3.4. Добавить ребра найденного остовного дерева в граф \tilde{G} :
 $E(\tilde{G}) := E(\tilde{G}) \cup E(T(\mathcal{T}))$, $B := B \cup E(T(\mathcal{T}))$.

EndFor

End of Algorithm

OE-покрытие несвязного графа



Теорема 5

Если в каждой компоненте связности G_k графа G степени вершин, инцидентных разделяющим граням графа G , четны, то маршрут с минимальной длиной дополнительных построений реализуется алгоритмом DoubleBridging.

5

⁵Макаровских Т.А., Панюков А.В., Савицкий Е.А. Математические модели и алгоритмы маршрутизации для САПР технологической подготовки процессов раскроя// Автоматика и телемеханика. 2017. № 5. С. 123–140.

- Рассмотрим эйлерову цепь

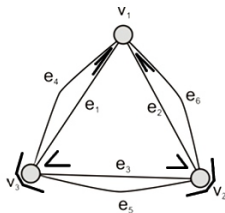
$$T = v_0, k_1, v_1, \dots, k_n, v_n, v_n = v_0$$

в графе $G = (V, E)$.

- Пусть в каждой вершине $v \in V$ задан циклический порядок $O^\pm(v)$, определяющий систему переходов $A_G(v) \subset O^\pm(v)$ в этой вершине.
- Когда $\forall v \in V(G) A_G(v) = O^\pm(v)$, систему переходов $A_G(v)$ будем называть **полной системой переходов**.

Определение [Г. Фляйшнер, 1990]

Цепь T будем называть **A -цепью**, если она является A_G -совместимой цепью, т.е. последовательные ребра в цепи T (инцидентные вершине v) являются соседями в циклическом порядке $O^\pm(v)$.

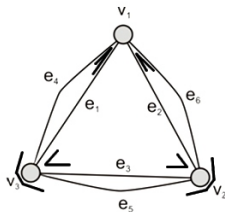


Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_6 v_2 e_5 v_3 e_4 v_1$ является A -цепью.

Цикл $v_1 e_1 v_2 e_3 v_3 e_2 v_1 e_4 v_3 e_5 v_2 e_6 v_1$ НЕ является A -цепью, т.к. переходы $e_2 v_1 e_4$ и $e_6 v_1 e_1$ пересекаются.

Определение

Будем говорить, что цепь является AOE -цепью, если она одновременно является OE -цепью и A -цепью.

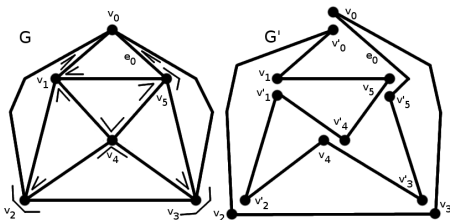


- $T_1 = e_1 e_3 e_2 e_4 e_5 e_6$ не является A -цепью, но является OE -цепью.
- Цепь $T_2 = e_1 e_3 e_2 e_6 e_5 e_4$ является AOE -цепью.
- A -цепь $T_3 = e_4 e_5 e_6 e_2 e_3 e_1$ не является OE -цепью.

Теорема 6

Если в плоском графе G графе существует A -цепь, то существует и АОЕ-цепь.

6



Плоский граф G с заданной в каждой вершине системой переходов для A -цепи и соответствующий ему граф G' , являющийся жордановой кривой на плоскости

⁶ *Makarovskikh T.A., Panyukov A.V.* АОЕ-Trails Constructing for a Plane Connected 4-Regular Graph. // CEUR Workshop Proceedings. Vol 1623. Pp. 62–71. Online: <http://ceur-ws.org/Vol-1623>

Теорема 7

В плоском связном 4-регулярном графе G существует *AOE*-цепь.

7

⁷ *Makarovskikh T.A., Panyukov A.V.* AOE-Trails Constructing for a Plane Connected 4-Regular Graph. // CEUR Workshop Proceedings. Vol 1623. Pp. 62–71. Online: <http://ceur-ws.org/Vol-1623>

Определение

Будем называть суграф G_k графа G такой, что

$$E(G_k) = \{e \in E(G) : \text{rank}(e) \geq k\},$$

суграфом ранга k .

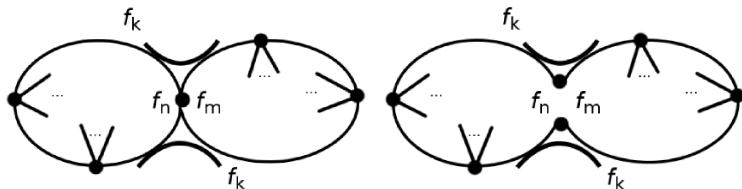
Предложение 1

Вершина, инцидентная четырем ребрам, смежным внешней грани, является точкой сочленения.

Предложение 2

Внешняя грань суграфа G_k является объединением всех граней ранга k в графе G .

Расщепление точек сочленения в суграфах ранга k



Теорема 8

Алгоритм CUT-POINT-SPLITTING определяет точки сочленения всех рангов плоского связного 4-регулярного графа и расщепляет их без нарушения связности за время не превосходящее $O(|E|)$.

8

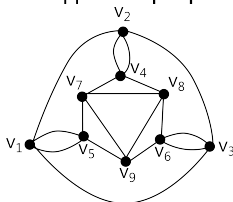
⁸Макаровских Т.А. Программное обеспечение для построения А-цепей с упорядоченным охватыванием в плоском связном 4-регулярном графе // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». 2019. Т.8. №1. С. 36–53.

АЛГОРИТМ CUT-POINT-SPLITTING. Данные

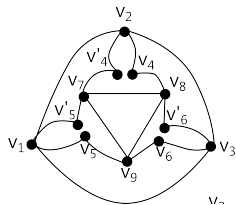
Входные данные: плоский связный 4-регулярный граф $G = (V, E)$, представленный для всех $e \in E(G)$ функциями $v_s, l_s, r_s, s = 1, 2$.

Выходные данные: гомеоморфный образ графа $G = (V, E)$, представленный для всех $e \in E(G)$ функциями $v_s, l_s, r_s, s = 1, 2$, в котором в любом суграфе G_k точки сочленения расщеплены.

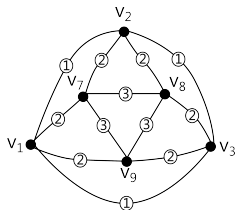
Исходный граф G



Граф \tilde{G} с расщепленными точками
сочленения



Гомеоморфный образ графа \tilde{G}



Алгоритм *AOE-TRAIL*

Input:

плоский связный 4-регулярный граф $G = (V, E)$ без точек сочленения любого ранга, представленный функциями $v_k, l_k, r_k, k = 1, 2$;
начальная вершина $v \in V(f_0)$.

Output:

ATrail – выходной поток,
содержащий построенную алгоритмом *AOE*-цепь.

Initiate(G, v_0); // Инициализация

Ranking(G); // Ранжирование

Constructing(); // Построение

Конец Алгоритма

Процедура Constructing

$e = \arg \max_{e \in E(v)} \text{rank}(e);$

$v = v_1(e);$

do {

if ($v \neq v_1(e)$) **REPLACE**(e);

$A\text{Trail} \ll v \ll e;$

$\text{mark}(e) = \mathbf{false};$ counter++; $v = v_2(e);$

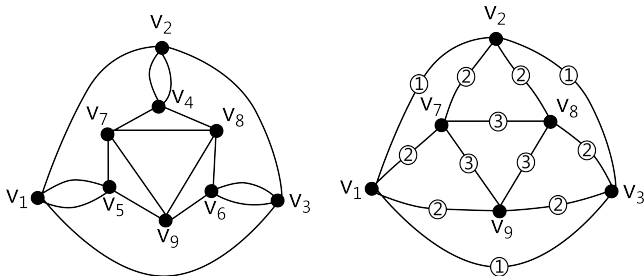
if ($\text{rank}(r_2(e)) \geq \text{rank}(l_2(e))$) **then**

if $\text{mark}(r_2(e))$ **then** $e = r_2(e)$ **else** $e = l_2(e);$

else if $\text{mark}(l_2(e))$ **then** $e = l_2(e)$ **else** $e = r_2(e);$

} **while**(counter < $|E(G)|$);

Конец Процедуры



$V_1 V_7 V_9 V_8 V_7 V_2 V_8 V_3 V_9 V_1 V_3 V_2 V_1,$

что соответствует цепи

$V_1 \mathbf{V_5} V_7 V_9 V_8 V_7 \mathbf{V_4} V_2 \mathbf{V_4} V_8 \mathbf{V_6} V_3 \mathbf{V_6} V_9 \mathbf{V_5} V_1 V_3 V_2 V_1$

в исходном графе.

Теорема 9

Алгоритм AOE -TRAIL строит AOE -цепь в плоском связном 4-регулярном графе G , любой суграф G_k , $k = 1, 2, \dots$ которого не содержит точек сочленения. Алгоритм находит решение за время $O(|E(G)| \cdot \log |V(G)|)$.

9

⁹Макаровских Т.А. Программное обеспечение для построения А-цепей с упорядоченным охватыванием в плоском связном 4-регулярном графе // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». 2019. Т.8. №1. С. 36–53.

Произвольный плоский граф $G: \forall v \in V(G), \deg(v) \leq 4$

Определение

AOE-покрытие – это OE-покрытие, каждая цепь, входящая в которое, соответствует системе переходов $A_G(v) \subset O^\pm(v)$.

Граф G дополнить граф до эйлера G^* введением вспомогательных ребер.

- Если вершина v – начало цепи, то первое ребро новой цепи (достигнута по дополнительному ребру) – ребро, инцидентное вершине максимального ранга.
- Если вершина v достигнута по ребру e графа, то переход осуществляется по ребру (основному либо дополнительному), инцидентному вершине максимального ранга. Если этот переход был осуществлен по дополнительному ребру, то вершина v является концом текущей цепи, а ребро e – последним в текущей цепи.

Получим 4-регулярный граф, гомеоморфный образ которого будет иметь на одну вершину меньше.

При таком построении все цепи будут A_G -совместимыми.

Определение

Эйлеров цикл в C плоском графе G называется самонепересекающимся, если он гомеоморфен циклическому графу \tilde{G} , который может быть получен из графа G с помощью применения $O(|E(G)|)$ операций расщепления вершин.

Определение

Будем говорить, что цепь является *NOE*-цепью, если она одновременно является *OE*-цепью и самонепересекающейся цепью.

Алгоритм *NOE-CHAIN* (G)

Вход: плоский эйлеров граф G , заданный функциями $v_k(e)$, $l_k(e)$, $r_k(e)$, $f_k(e)$, $k = 1, 2$ и $\text{rank}(e)$;

Выход: *NOE*-цепь в графе G ;

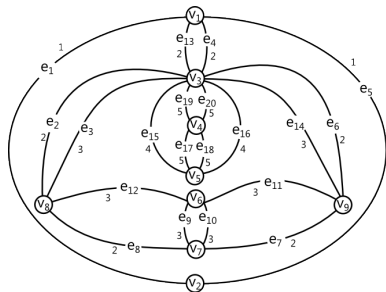
$\hat{G} = \text{NonIntersecting}(G)$; // Расщепить все вершины степени выше 4

$\tilde{G} = \text{CutPointSplitting}(\hat{G})$; // Расщепить все точки сочленения всех рангов

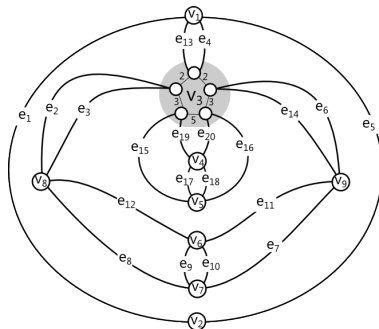
$C^* = \text{AOE_TRAIL}(\tilde{G})$; // Построить *AOE*-цепь в графе \tilde{G}

$C = \text{Absorb}(C^*)$; // Стянуть все расщепленные вершины

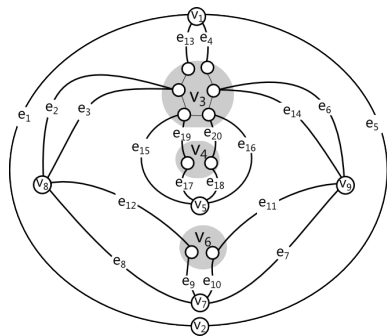
Пример использования алгоритма *NOE-CHAIN* (G)



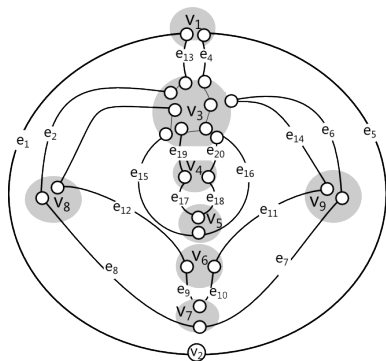
а) Граф G



б) 4-регулярный граф \hat{G}



в) 4-регулярный граф \tilde{G} без точек сочленения



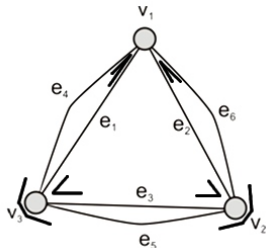
г) самонепересекающийся *OE*-маршрут C

- Для плоского неэйлерова (в общем случае не связного) графа G расщепить все вершины степени выше 4 . Получим граф, степени вершин которого равны 3 или 4.
- Для этого графа применить алгоритм построения AOE -покрытия.
- В цепях полученного покрытия удалить все искусственные ребра и стянуть все расщепленные вершины.
- В результате получим NOE -покрытие.

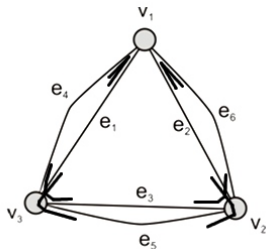
Предложение 1

Пусть $G(V, E)$ – плоский эйлеров граф без точек сочленения и представляет OE -цепь в графе G , которой соответствует система переходов $X_T(G)$. Тогда число OE -цепей N для системы переходов $X_T(G)$ удовлетворяет неравенству $1 \leq N \leq 2 \cdot |V(f_0)|$, $V(f_0) = \{v \mid v \in f_0\}$, причем как верхняя, так и нижняя оценки достижимы.

Задана система переходов



- $C_1 = v_1 e_1 v_3 e_3 v_2 e_2 v_1 e_6 v_2 e_5 v_3 e_4 v_1$ – ОЕ-цепь;
- $\tilde{C}_1 = v_1 e_6 v_2 e_5 v_3 e_6 v_1 e_1 v_3 e_3 v_2 e_2 v_1$, в ней цикл $v_1 e_6 v_2 e_5 v_3 e_6 v_1$ охватывает еще не пройденные ребра $v_1 e_1 v_3 e_3 v_2 e_2 v_1$;
- $C_{1,2} = v_2 e_5 v_3 e_4 v_1 e_1 v_3 e_3 v_2 e_2 v_1 e_6 v_2$ – ОЕ-цепь;
- $C_{1,3} = v_3 e_4 v_1 e_1 v_3 e_3 v_2 e_2 v_1 e_6 v_2 e_5 v_3$ – ОЕ-цепь.



В этом графе существует единственная ОЕ-цепь

$$C = v_2 e_3 v_3 e_4 v_1 e_1 v_3 e_5 v_2 e_2 v_1 e_6 v_2,$$

удовлетворяющая заданной системе переходов $X_C(G)$.

Теорема 10

Пусть плоский граф $G = (V, E)$ без разделяющих вершин имеет A -цепь T , которой соответствует система переходов $X_T(G)$. Если $V(f_0)$ – множество вершин, смежных внешней грани, то число ОЕ-циклов для $X_T(G)$ равно $2 \cdot |V(f_0)|$.

10

¹⁰ Макаровских Т.А. О числе ОЕ-цепей для заданной системы переходов// Вестник Южно-Уральского государственного университета. Серия «Математика. Механика. Физика». 2016. Т.8. №1. С. 5–12.

Теорема 11

Пусть плоский граф $G = (V, E)$ имеет K разделяющих вершин $v_1, \dots, v_K \in f_0$ и пусть в этом графе существует А-цепь T . Пусть $X_T(G)$ – система переходов, соответствующая T , а $V(f_0)$ – множество вершин, смежных внешней грани. Существует

$$2 \cdot |V(f_0)| + \sum_{i=1}^K (\deg(v_i) - 2)$$

ОЕ-циклов для $X_T(G)$.

11

¹¹Макаровских Т.А. О числе ОЕ-цепей для заданной системы переходов// Вестник Южно-Уральского государственного университета. Серия «Математика. Механика. Физика». 2016. Т.8. №1. С. 5–12.

- 1 Введены классы маршрутов в плоских графах, удовлетворяющих различным наборам технологических ограничений:
 - маршруты с упорядоченным охватыванием (OE),
 - самонепересекающиеся OE-маршруты (NOE).

Доказано существование маршрутов для классов OE и NOE.

- 2 Определен способ представления гомеоморфного образа раскройного плана, позволяющего эффективно решать поставленные проблемы маршрутизации:
 - разработаны полиномиальные алгоритмы решения проблемы построения маршрутов указанного вида в плоских графах;
 - доказана их корректность;
 - разработано программное обеспечение для реализации представленных алгоритмов.
- 3 Получены оценки количества полученных маршрутов специального вида.
- 4 Решены практические задачи, предложенные кафедрой информационных технологий и автоматизации проектирования УрФУ.

СПАСИБО ЗА ВНИМАНИЕ!