AOE-Trails Constructing for a Plane Connected 4-Regular Graph

Tatiana Makarovskikh and Anatoly Panyukov

South Ural State University,
pr. Lenina, 76, 454080 Chelyabinsk, Russia
Makarovskikh.T.A@susu.ru
paniukovav@susu.ru

Abstract. The polynomial algorithm for constructing a special Eulerian cycle in a plane connected 4-regular graph $G = (V, E)$ is presented in the paper. First of all any cycle of the passed edges does not contain any unpassed edges (the condition of ordered enclosing (OE-cycle). At second, the next edge of this cycle can be chosen from one of two neighbouring edges (left or right) of a cyclic order for the end-vertex of the current edge (the A-trail condition). The computing complexity of this algorithm is $O(|E(G)| \log |V(G)|)$.

Keywords: 4-regular graph, A-trail, ordered enclosing, algorithm

Introduction

It is known that with the help of Leonhard Euler's theorem we can easily determine whether a given graph has Eulerian cycle. In order Eulerian cycle to exist in the graph it is necessary and sufficient that all the vertices of this graph have even degree. However, the problem of constructing the Eulerian cycle with some restrictions is not too trivial. For example, for the problem which requires that in a planar Eulerian graph the cycle of passed edges does not cover unpassed edges (the problem of constructing the OE-trail) in spite of the polynomial solvability requires a special data organization [6]. In common constructing of trail satisfying the given transitions system is $\mathcal{NP}$-hard problem [3]. Its partial case known as a problem of A-trail constructing when transition system for each vertex be a cycle remains $\mathcal{NP}$-hard. Herbert Fleischner determined some classes of graphs for which the polynomail algorithms of finding the A-trail exist. For example, in [1] this problem is solved for outerplanar graphs.

In this paper we consider the case of connected 4-regular plane graphs. The existence of polynomial time algorithm for constructing the A-trail C being the OE-trail [5] is proved. This kind of trail is called as AOE-trail.

The considered problem has its implementation for cutting processes management if it is necessary to take into account some technological restrictions for cutter trajectory. Here the homeomorphic image of a cutting plan be a plane graph usually without separating vertices. For example, the restrictions used for constructing the OE-trail

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

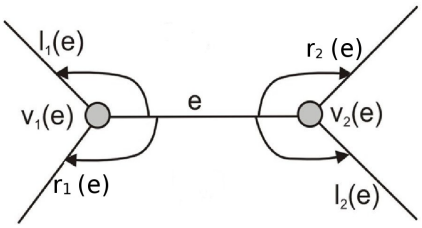

Fig. 1. Functions representing a plane graph

can be interpreted as a claim to avoid the cuts of a part cut off a sheet. The sequence of details cutting is not important in this case. Nevertheless, it is not so in practice. For example, using flame cutting technology we need absence of cuts intersections. As the model of this problem solution we can consider an *A*-trail. So, *AOE*-trail allows define such a trajectory of cutter that avoids additional cuttings of part cut off a sheet and the next cut off fragment is a neighbouring one.

1 Definitions and Designations

Let for plane graph G the set $E(G)$ be a set of its edges being the plane Jordan curves with mutually intersecting entrails homeomorphic to segments. Edge with the same starting and ending point be called as a loop of graph. Let $V(G)$ be a set of bounding points of these curves. Topological representation of plane graph $G = (V, E)$ on plane S up to homeomorphism can be determined by the following functions for each edge $e \in E$ [5]:

- $v_k(e)$, $k = 1, 2$ be vertices incident to edge e,
- $l_k(e)$, $k = 1, 2$ be edges got by rotating edge e counter clockwise around vertex $v(k)$,
- $r_k(e)$, $k = 1, 2$ be edges got by rotating edge e clockwise around vertex $v(k)$.

The illustration of these functions is given on figure 1. Constracting of these functions has no problems. In fact they are defined and used on the stage of graph G design. The space complexity of this representation be $O(|E| \cdot \log_2 |V|)$.

Definition 1. ***Graph of allowed transitions*** *[3] (or shortly,* ***transitions graph****)* $T_G(v)$ *for vertex* $v \in V(G)$ *be a graph which vertices be the edges incident to vertex* v*, i.e.* $V(T_G(v)) = E_G(v)$*, and the set of edges be represented by allowed transitions between edges.*

Definition 2. ***System of allowed transitions*** *[2] (or shortly,* ***transitions system****)* T_G *be a set*

$$\{T_G(v) \mid v \in V(G)\}$$

where $T_G(v)$*be transitions graph of vertex* v*.*

Definition 3. *Let $P = v_0, e_1, v_1, ..., e_k, v_k$ for graph G be T_G-**compatible** if $e_i,\ e_{i+1} \in E(T_G(v_i))$ for each i $(1 \le i \le k-1)$.*

The following definition is given in [2].

Definition 4. *Let for Eulerian trail*

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

*in graph $G = (V, E)$ the cyclic order $O^{\pm}(v)$ is given for each vertex $v \in V$. This order defines the transitions system $A_G(v) \subset O^{\pm}(v)$. If $\forall v \in V(G)\ A_G(v) = O^{\pm}(v)$ let transitions system $A_G(v)$ be called **full transitions system**.*

Definition 5. *Eulerian A_G-compatible trail T be called A-**trail** Hence the sequent edges of trail T (incident to vertex v) be the neighbours in cyclic order $O^{\pm}(v)$.*

As it was mentioned above there are some classes of graphs [2] for which the recognition of A-trail existence claims polynomial time. Algorithms for outer-planar graphs [1] and 4-regular graphs [2] are also known. In [2, Corollary VI.6] the proof that A-trail exists for any connected 4-regular graph on any surface is considered. To prove this fact author uses the Splitting lemma. It is also proved that there exists polynomial algorithm for recognition of A-trail for a 4-regular graph.

Let's consider the plane S. Let plane Eulerian graph $G = (V, E)$ be set on this plane. Let f_0 be the outer face of graph G. Let's define $\mathrm{Int}(H)$ for any subset $H \subset S$ as subset of S being the union of all components of set $S \backslash H$ without outer face f_0. So, $\mathrm{Int}(G)$ be the union of all inner faces of graph G.

We take the neighbours of edge e got after rotating this edge clockwise and counter clockwise in vertex v as sequence of edges in $O^{\pm}(v)$ for plane graph.

Let's consider graph on figure 2. The bold arcs show the allowed transitions of this graph.

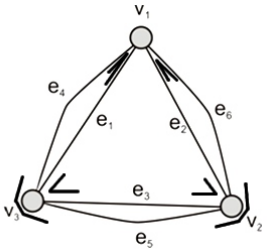

Fig. 2. Example of Eulerian graph

Let's define the condition of ordered enclosing [5] for plane Eulerian graph.

Definition 6. *We say that cycle $C = v_1e_1v_2e_2\dots v_k$ of Eulerian graph G has ordered enclosing (or being OE-trail for short) if for any its initial part $C_i = v_1e_1v_2e_2\dots e_i$, $i \le |E(G)|$ condition*

$$\mathrm{Int}\,(C_i) \cap G = \emptyset$$

holds.

Definition 7. *We say a trail to be AOE-trail if it is OE-trail and A-trail simultaneously.*

Let's consider trail $T_1 = e_1e_3e_2e_4e_5e_6$ of graph on figure 2. This trail is not A-trail but it is OE-trail. Trail $T_2 = e_1e_3e_2e_6e_5e_4$ is to be AOE-trail. For example, A-trail $T_3 = e_6e_5e_4e_1e_3e_2$ is not OE-trail.

We need the following definition for the further considerations and proofs.

Definition 8. *Let rank of edge $e \in E(G)$ be called a value of function* rank(e) : E(G) $\to \mathbb{N}$ *recursively defined:*

- *let $E_1 = \{e \in E : e \subset f_0\}$ be a set of edged bounding outer face f_0 of graph $G(V, E)$ then $(\forall e \in E_1)\,(\mathrm{rank}(e) = 1)$;*
- *let $E_k(G)$ be a set of edges having rank = 1*

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

then $(\forall e \in E_k)\,(\mathrm{rank}(e) = k)$.

Definition 9. *Partial graph G_k of graph G that $E(G_k) = \{e \in E(G) :$ rank(e) $\ge$ k$\}$ be called a* ***partial graph of rank*** *k.*

Definition 10. *The* ***rank of vertex*** *$v \in V(G)$ be the value of function* rank : $V(G) \to$ N*:* $\mathrm{rank}(v) = \min_{e\in E(v)} \mathrm{rank}(e)$ *where $E(v)$ be the set of edges incident to vertex $v \in V$.*

Definition 11. *The* ***rank of face*** *$f \in F(G)$ be the value of function* rank : $F(G) \to$ $\mathrm{Z}^{\ge 0}$*:*

$$\mathrm{rank}(f) = \begin{cases} 0, & \text{if } f = f_0, \\ \min_{e\in E(f)} \mathrm{rank}(e), & \text{otherwise} \end{cases}$$

where $E(f)$ be the set of edges incident to outer face $f \in F$.

The different ways of rank(e) function defining are given in papers [5], [7]. Computing complexity of definition the rank for all graph edges is less than $O(|E| \log_2 |V|)$.

2 Existence of Transition System Allowing the *AOE*-Trail

A trail for which the condition of ordered enclosing is vanished always has a transition through the enclosing cycle. This transition is not compatible with the transitions system for A-trail. On the other hand the following theorem holds.

Theorem 1. *If there is an A-trail for a plane graph G then there is also an AOE-trail.*

Proof. *A*-trail for a plane Eulerian graph is to be the closed Jordan curve without self-intersections. This curve may be got by the following way. Let's consider graph G, there is a transition system for each vertex corresponding the *A*-trail. Let graph G' is obtained of graph G by splitting the vertices according to the system of allowed transitions (figure 3). Graph G' obtained this way represents a Jordan curve without intersections and according to Jordan theorem this curve splits the plane into outer and inner components.

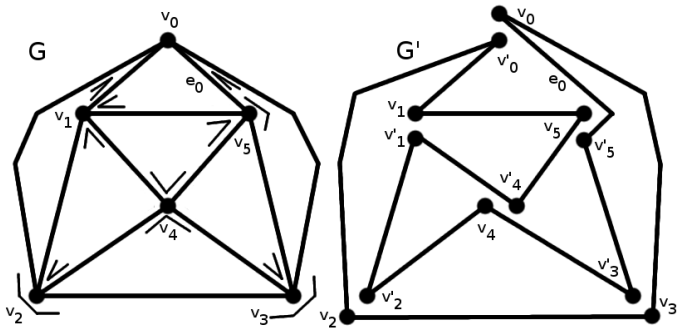

Fig. 3. Plane graph G (each its vertex has a defined transitions system for *A*-trail) and corresponding graph G' being the Jordan curve on a plane

Obviously, there exists vertex v_0 on the curve obtained incident to outer face of graph G and to edge e_n: $\mathrm{rank}(e_n) = 1$. If v_0 be taken as beginning of *AOE*-trail and the direction of passing is chosen so that edge (e_n) be the end of trail C_n than due to absence of intersections for the inner set of any initial part of a trail $C_i = v_0e_1v_1e_2 \dots e_i$, $i \leq |E(G)|$ the condition

$$\mathrm{Int}\,(C_i) \cap G = \emptyset$$

holds, i.e. such an *A*-trail C_n be also an *OE*-trail. Really, if we consider the existence of edge $e \in \mathrm{Int}\,(C_i)$ then we get a contradiction to absence of intersections in transition system.

Theorem 2. *There is an AOE-trail for plane connected 4-regular graph G.*

Proof. The proof of *A*-trail existence for a connected 4-regular graph is given in [2, Corollary VI.6]. As this graph contains an *A*-trail then due to theorem 1 there is also an *AOE*-trail.

3 Algorithm for *AOE*-Trail Constructing

Let's consider the algorithm for constructing *AOE*-trail for plane connected 4-regular graph. Let's input and output data be the global variables for shortness of exposition.

ALGORITHM *AOE*-**TRAIL**

Input: plane connected 4-regular graph $G = (V, E)$ without cut-vertices for partial graph G_k, $k = 1, 2, \ldots$ represented for all $e \in E(G)$ by functions v_s, l_s, r_s , $s = 1, 2$ (figure 1);

$v \in V(f_0)$ be starting vertex.

Output:

$ATrail$ be the output-stream,

containing the AOE-trail constructed by algorithm.

Initiate(G)**;**

Ranking(G)**;**

Constructing();

End of algorithm

The (`Initiate()`) procedure initializes by value 0 the `counter` of edges included to a resulting trail, and also assigning to all edges the mark `true` corresponding the unpassed edge.

The functional purpose of the procedure `Ranking()` is to define rank(e) for each graph edge $e \in E(G)$ and rank(v) for each graph vertex $v \in V(G)$. As it was mentioned above the different algorithms of defining rank(e) function are given in papers [5], [7], and their computing complexity is less than $O(|E| \log_2 |V|)$.

The description of `Constructing()` procedure is given below.

Constructing ()

$e = \arg\max_{e \in E(v)} \text{rank}(e)$;
$v = v_1(e)$;
for counter=1 **to** $|E(G)|$ **do** {
  if $(v \neq v_1(e))$ REPLACE(e);
  $ATrail$ << v << e;
  mark(e) = **false**; counter++; $v = v_2(e)$;
  if (rank($r_2(e)$) > rank($l_2(e)$)) **then**
    if mark($r_2(e)$) **then** $e = r_2(e)$ **else** $e = l_2(e)$;
  else if mark($l_2(e)$) **then** $e = l_2(e)$ **else** $e = r_2(e)$;
  }

End of Procedure

While running procedure **Constructing** (e) the interchange of numbers of incident to e vertices appear, so that vertex $v_1(e)$ should be passed earlier than vertex $v_2(e)$. This procedure is realized as the following.

Procedure REPLACE (*Edge* **)**

$tmp1 = v_2(Edge)$; $tmp2 = l_2(Edge)$; $tmp3 = r_2(Edge)$;
$v_2(Edge) = v_1(Edge)$; $l_2(Edge) = l_1(Edge)$; $r_2(Edge) = r_1(Edge)$;
$v_1(Edge) = tmp1$; $l_2(Edge) = tmp2$; $r_2(Edge) = tmp3$;

End of Procedure

The following theorem holds.

Theorem 3. *Algorithm AOE-TRAIL constructs AOE-trail for a plane connected 4-regular graph G so that any its partial graph G_k, $k = 1, 2, \ldots$ does not contain cut-vertices. Algorithm runs by time $O(|E(G)| \cdot \log |V(G)|)$.*

Proof. Effectiveness of procedures `Initiate()` and `Ranking()` is obvious and their total computing complexity is less than $O(|E(G)|\log|V(G)|)$.

The main cycle of procedure `Constructing()` is in choosing the next unpassed edge e incident to vertex $v_2(e)$ maximal rank. This procedure is run until all the edges are not included to the resulting trail (their mark will be changed to `false`).

Let's prove the effectiveness of `Constructing()` procedure. If for the current edge e its vertex $v = v_2(e)$ is visited for the first time then running of body of the cycle may be interpreted as splitting the vertex $v^* = v_2(e)$ according to the system of transitions of A-trail.

After such a splitting the homeomorphic image of the graph obtained does not contain the vertex v^*. On the second visit to vertex $v^* \in V(G)$ algorithm continues forming the trail according to homeomorphic image of graph obtained earlier.

The homeomorphic image of obtained graph be a plane connected graph after each running of the cycle body since any partial graph of rank k contains no cut-vertices.

After running of $|V(G)|$ splits the resulting homeomorphic image of graph be a circle obtained by splitting the vertices according to the transitions system of A-trail. Wherein the stream $ATrail$ contains the resulting AOE-trail.

As homeomorphic image remains connected then all edges are treated by algorithm (i.e. have mark `false`). Procedure `Constructing()` has the only cycle. Computing complexity of cycle body is less than $O(\log|V(G)|)$ and the number of iterations is equal to $|E(G)|$. Hence, computing complexity of algorithm is less than $O(|E(G)|\log|V(G)|)$.

The algorithm claims rather strict conditions on the graph on absence of cut-vertices for all partial graphs G_k. However, if we "properly" fix the transitions for all these cut-vertices then as the result of splitting we get a graph for which any partial graph does not contain any cut-vertices. The correct transition is one between the arcs satisfying cyclic transition system and incident to different pairs of faces (see figure 4). Searching

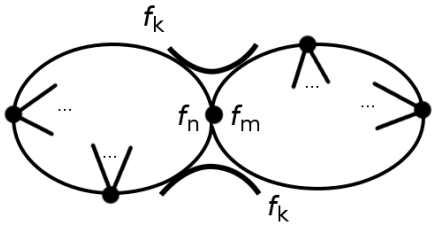

Fig. 4. The correct transition system splitting the cut-vertex in partial graph G_k

of cut-vertices uses the following propertioes of 4-regular graphs.

Proposition 1. *Vertex incidence to four edges bounding outer face is cut-vertex.*

Proposition 2. *Outer face of G_k is a union of all faces of rank k for graph G.*

These statements truth are obvious.

All this allows to suggest the following algorithm.

ALGORITHM CUT-POINT-SPLITTING

Input: plane connected 4-regular graph $G = (V, E)$ represented for all $e \in E(G)$ by functions v_s, l_s, r_s , $s = 1, 2$ (figure 1).

Output: homeomorphic image of graph $G = (V, E)$ with splitted cut-vertices represented for all $e \in E(G)$ by functions v_s, l_s, r_s , $s = 1, 2$.

Variables: $\forall v \in V(G)$: arrays point(v), rank(v), count(v); $\forall f \in F(G)$: array rank(f).

Initiate():
for $\forall v \in V(G)$ point$(v) = 0$; count$(v) = 0$;
Ranking(G); //Function counts ranks of all vertices, edges, and faces
Finding():
for $\forall e \in E(G)$ **do**
    point$(v_1(e)) = e$; point$(v_2(e)) = e$;
enddo
for $\forall v \in V(G)$ **do**
    $e =$ point(v); $k =$ rank(v);
    if $(v = v_1(e))$ **then** $s = 1$ **else** $s = 2$;
    $e = l_s(e)$;
    if (rank$(e) = k$) **then** count$(v) =$ count$(v) + 1$, $e = l_s(e)$;
    if (rank$(e) = k$) **then** count$(v) =$ count$(v) + 1$, $e = l_s(e)$;
    if (rank$(e) = k$) **then** count$(v) =$ count$(v) + 1$,
    if (count$(v) = 4$) **then**
        if $((f_s(e) = f_s(l_s(e))$ **and** $f_{3-s}(e) = f_{3-s}(l_s(e)))$ **or**
                $(f_s(e) = f_{3-s}(l_s(e))$ **and** $f_{3-s}(e) = f_s(l_s(e))))$ **then**
            $e^* = l_s(e)$, $l_s(e) = r_s(e)$, $r_s(r_s(e)) = e$,
            $r_s(e^*) = l_s(e^*)$, $l_s(l_s(e^*)) = e^*$;
        else
            $e^* = r_s(e)$, $r_s(e) = l_s(e)$, $l_s(l_s(e) = e$,
            $l_s(e^*) = r_s(e^*)$, $r_s(r_s(e^*)) = e^*$;
endfor
End of algorithm

Computing complexity of this algorithm is $O(|E(G)| \log |V(G)|)$.

Let's consider the algorithm on example (figure 5).

After running `CUT-POINT-SPLITTING` algorithm we have graph presented on figure 6.

Figure 7 shows the homeomorphic image of graph on figure 6.

The *AOE*-trail starting at vertex v_1 is

$$v_1 v_7 v_9 v_8 v_7 v_2 v_8 v_3 v_9 v_1 v_3 v_2 v_1$$

which corresponds the trail

$$v_1 \mathbf{v_5} v_7 v_9 v_8 v_7 \mathbf{v_4} v_2 \mathbf{v_4} v_8 \mathbf{v_6} v_3 \mathbf{v_6} v_9 \mathbf{v_5} v_1 v_3 v_2 v_1$$

in initial graph.

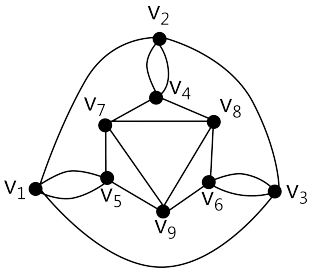

Fig. 5. Example of plane 4-regular graph

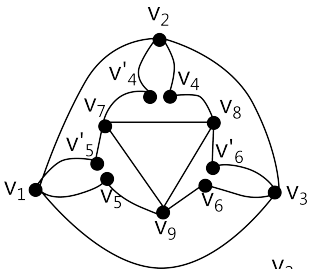

Fig. 6. Graph with splitted cut-vertices

Conclusion

Hence, any A-trail for a plane graph up to choosing the start vertex and routing direction is to be an AOE-trail. It is possible to construct Eulerian cycle being AOE-trail for plane 4-regular graph by presented algorithm. For graphs with vertices of degree greater than 4 this algorithm may not construct any AOE-trail. Further interest are algorithms for constructing AOE-trails for arbitrary plane connected Eulerian graph.

Acknowledgments. The authors thank professor of TU Vienna Herbert Fleischner for statement and productive discussion of the problem.

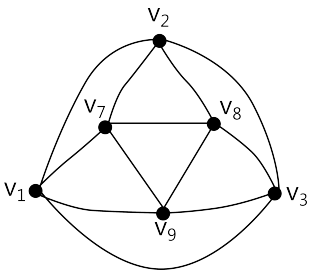

Fig. 7. Homeomorphic image of graph on fig.6

References

1. Andersen, L.D., Fleischner, H., Regner, S.: Algorithms and outerplanar conditions for *A*-trails in plane Eulerian graphs. Discrete Applied Mathematics, no.85, 99–112 (1998).
2. Fleischner, H. Eulerian Graphs and Related Topics, Part 1, Vol.1, Ann. Discrete Mathematics, No 45 (1990).
3. Szeider, S. Finding Paths in Graphs Avoiding Forbidden Transitions. Discrete Applied Mathematics. No 126. 261–273 (2003).
4. Makarovskikh, T.A. Covering the rectangular cutting plan by AOE-trails. In: Infornation Technologies and Systems: Proc. of 4-th International conference. Chelyabinsk. Pp. 17–18. (2015) (in Russian).
5. Panyukova, T.A. Trails with ordered enclosing for plane graphs. Discrete Analysis and Operation Research. Part 2. Vol.13, No 2, Pp. 31–43 (2006) (in Russian).
6. Panyukova, T.A. Optimal Eulerian covers for plane graphs. Discrete Analysis and Operation Research. Vol. 18, No 2. Pp. 64–74 (2011) (in Russian).
7. Savitskiy, E.A. Using the Breadth-first Search Algorithm for Definition of Plane Graph Edges Ranks. In: Infornation Technologies and Systems: Proc. of 3-rd International conference. Chelyabinsk. Pp.43–45 (2014) (in Russian).