

## THE SOFTWARE FOR ALGORITHMS OF ORDERED ENCLOSING COVERING CONSTRUCTING FOR PLANE GRAPHS

T. PANYUKOVA<sup>1</sup>, E. SAVITSKIY<sup>2</sup>

<sup>1</sup>kwark@mail.ru, <sup>2</sup>egor88@inbox.ru  
 South Ural State University, Chelyabinsk, Russian Federation

*Submitted 2013, June 10*

**Abstract.** This paper represents a review of software for constructing ordered enclosing covering for a plane graph with any number of connected components. The authors present the effective algorithm for optimal solution of the considered problem.

**Keywords:** cutting process; planar graph; Eulerian trail; algorithm.

### 1. INTRODUCTION

Let's the finite number of details is laid on a sheet. The problem is to define the path of cutter movement so that part cut off from a sheet does not require additional cuttings. To formalize this restriction let's introduce the definition of trails sequences with ordered enclosing [1].

As a model of a sheet lets assume the plane  $S$ , as a model of cutting plan let's assume the plane graph  $G$  with interior facet  $f_0$  on plane  $S$ . Let  $J$  be a part of cutter trajectory. As  $\text{Int}(J)$  let's designate the union of all connected components  $S \setminus J$  without exterior facet. Then  $\text{Int}(J)$  be a part cut off from a sheet.

**Definition 1.** The cycle  $C = v_1 e_1 v_2 e_2, \dots, v_k$  in an Eulerian graph  $G$  is said to have an ordered enclosing, if for any initial part  $C_l = v_1 e_1 v_2 e_2, \dots, v_l$ ,  $l \leq |E|$ , the condition  $\text{Int}(C_l) \cap E = \emptyset$  is satisfied [1].

**Definition 2.** We say that minimal cardinality sequence of edge disjoint trails

$$C^0 = v^0 e_1^0 v_1^0 e_2^0 \dots e_{k_0}^0 v_{k_0}^0, C^1 = v^1 e_1^1 v_1^1 e_2^1 \dots e_{k_1}^1 v_{k_1}^1, \dots,$$

$$C^{n-1} = v^{n-1} e_1^{n-1} v_1^{n-1} e_2^{n-1} \dots e_{k_{n-1}}^{n-1} v_{k_{n-1}}^{n-1}$$

with ordered enclosing such that

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

is Eulerian cover with ordered enclosing for plane graph  $G=(V, E)$  [2].

Constructing the cover with ordered enclosing for graph  $G$  the given cutting problem will be

solved. The most interesting are covers with minimal number of trails because transition from one trail to another corresponds idling of the cutter.

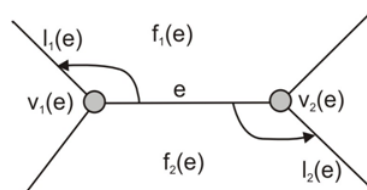
**Definition 3.** Let minimal cardinality sequence of edge-disjoint trails with ordered enclosing in plane graph  $G$  be called Eulerian cover with ordered enclosing [2].

### 2. REPRESENTATION OF CUTTING PLAN AS A GRAPH AND ALGORITHM SOLVING THE PROBLEM

As numerical control (NC) machines have programs designed for cutting many primitives that are the components of the curves outlining the pieces, we are interested in the order of enclosing of edges in graph  $G$ , rather than their specific form. Up to homeomorphism, any plane graph  $G=(V, E)$  can be represented by specifying for each edge  $e \in E$  the following functions (Fig. 1):

$v_1(e)$  and  $v_2(e)$  are the vertices incident to edge  $e$ ;

$f_k(e)$ ,  $k=1,2$  is the face located to the left when moving along the edge  $e$  from the vertex  $v_k(e)$  to the vertex  $v_{3-k}(e)$ ,  $k=1,2$ .



**Fig. 1.** Data representation of graph edges

$l_1(e)$  and  $l_2(e)$  are the edges belonging to the faces  $f_k(e)$ ,  $k=1,2$ , incident to the vertices  $v_k(e)$ .

These functions can be constructed without any problem. In fact, they are determined and used as early as at the stage of developing the cutting design (i. e., the graph  $G$ ). The computational complexity of this representation is  $O(|E|)$ .

The theorem of minimal cardinality Eulerian cover with ordered enclosing existence for plane graph  $G$  and algorithm of its constructing are cited in [2].

**Theorem 1.** *Let  $G=(V,E)$  be topological plane graph on  $S$  containing the set  $Odd(V(G))$  of  $2n$  odd degree vertexes, and  $f_0$  is exterior face of  $G$ .*

*If  $\exists v^* \in Odd(V(G)) \cap f_0$  than there exists Eulerian covering with ordered enclosing that consists or trails*

$$\begin{aligned} C^0 &= v^0 e_1^0 v_1^0 e_2^0 \dots e_{k_0}^0 v_{k_0}^0, \quad C^1 = v^1 e_1^1 v_1^1 e_2^1 \dots e_{k_1}^1 v_{k_1}^1, \dots \\ C^{n-1} &= v^{n-1} e_1^{n-1} v_1^{n-1} e_2^{n-1} \dots e_{k_{n-1}}^{n-1} v_{k_{n-1}}^{n-1}, \\ Odd(V(G)) &= \{v^* = v^0, v_{k_0}^0, \dots, v^{n-1}, v_{k_{n-1}}^{n-1}\}. \end{aligned}$$

*If  $V_{odd} \cap f_0 = \emptyset$  than there exists Eulerian covering with ordered enclosing that consists of trails*

$$\begin{aligned} C^0 &= v^0 e_1^0 v_1^0 e_2^0 \dots e_{k_0}^0 v_{k_0}^0, \quad C^1 = v^1 e_1^1 v_1^1 e_2^1 \dots e_{k_1}^1 v_{k_1}^1, \dots \\ C^{n-1} &= v^{n-1} e_1^{n-1} v_1^{n-1} e_2^{n-1} \dots e_{k_{n-1}}^{n-1} v_{k_{n-1}}^{n-1}, \quad C^n = v^* e_1^n v_1^n e_2^n \dots e_{k_n}^n v^*, \\ Odd(V(G)) &= \{v^0, v_{k_0}^0, \dots, v^{n-1}, v_{k_{n-1}}^{n-1}\}, \quad v^* \in f_0. \end{aligned}$$

The proof of this theorem is constructive and it shows that algorithm **OrderedEnclosingCover** for constructing cover by sequence of trails with ordered enclosing is effective [1].

To describe the work of algorithm we need a definition of nesting value for an edge  $e$ :  $kmark(e)$ .

**Definition 4.** Let the nesting value of edge  $e$  of plane topological graph  $G(V,E)$  be a value of function  $kmark(e)$  recursively defined as following:

All edges incident to exterior facet  $f_0$  of graph  $G(V,E)$  form a set of edged  $E_1 = \{e \in E : e \subset f_0\}$  with nesting value  $(\forall e \in E_1)(kmark(e) = 1)$ ;

Edges of nesting value equal to 1 in graph

$$G_k \left( V, E \setminus \left( \bigcup_{l=1}^{k-1} E_l \right) \right), \quad k \geq 2$$

form a set  $E_k$  of edges of nesting value  $k$  for graph  $G$ , i. e.  $(\forall e \in E_k) (kmark(e) = k)$ .

Algorithm **OrderedEnclosingCover** [4] for constructing Eulerian cover by trails with ordered enclosing consists of three consecutive procedures: «Initialization»; «Ordering»; «Forming». At **initial-**

**ization** stage all initial values are defined, the edge  $e_0 \in E$  belonging to boundary of external facet  $f_0$  is determined. Procedure «**Ordering**» is used for definition of function  $kmark(e)$  value for each edge  $e \in E$ , and for organization of the list of incident edges for each vertex. Vertices are sorted by decreasing of

$$kmark(v) = \min_{e: v \in e} kmark(e)$$

value. Function  $kmark(e)$  is used for definition of iteration number on which the edge  $e$  received its mark.

Prior to running «**Forming**» procedure let's sort vertices of odd degree by decreasing their  $kmark(v)$  value.

After sorting the edges by decreasing of their  $kmark(e)$  values the third stage of algorithm – **Forming** – begins. Here we construct  $|V_{odd}|/2$  of simple trails between pairs of vertices of odd degree as following:

Take the vertex with the greatest value of  $kmark(e)$ ;

Construct a trail from this vertex to another vertex of odd degree;

Delete the end-vertices from a list of odd degree vertices.

Using procedure «**Forming**» algorithm constructs maximal cardinality trail

$$\begin{aligned} C^l &= v_l e_1 v_2 e_2 \dots e_l v_{l+1}, \quad e_i = \arg \max_{e \in E(v_i) \setminus \{e_j : l < j\}} kmark(e), \\ v_{l+1} &= v_1(e_i), \end{aligned}$$

corresponding the condition of ordered enclosing.

As a result of this procedure running cycle  $C^n$  is received. It consists of edges adjacent exterior facet and also of trails with ordered enclosing  $C^i$ ,  $i = 1, 2, \dots, n-1$  where  $n = |V_{odd}|/2$ . Obviously, computing complexity of such algorithm is not more than  $O(|E| \cdot \log_2 |V|)$  operations.

This algorithm uses lexicographical ordering of odd degree vertices and allows to get any allowed solution (may be optimal, may be not) and it works only for a connected graph.

To optimize the length between the end of previous trail and beginning of the next one there can be used different algorithms. For example, greedy algorithm [3] allows to get not worse solution than one with lexicographical order.

## 2.1. Optimal Cover Algorithm

Let's consider the result proved in [4] and allowing to construct the set of trails with ordered

enclosing for any set  $M$  forming a matching on a set of odd degree vertices  $V_{odd}$ .

**Theorem 2.** Let  $G=(V,E)$  be a topological planar graph on a plane  $S$ . This graph does not have any vertices of degree 1. For any set  $M$  being a matching on the set of  $V_{odd}$  (it consists of  $2n$  odd degree vertices of graph  $G$ ) so that  $M:(M \cap S) \setminus V = \emptyset$  there exists Eulerian cycle  $C = v_1 e_1 v_2 e_2 \dots e_n v_1$ ,  $n = |E| + |M|$  for any initial part of which  $C = v_1 e_1 v_2 e_2 \dots v_l$ ,  $l \leq |E| + |M|$  the condition  $\text{Int}(C_l) \cap E = \emptyset$  is held.

Let's consider algorithm `OptimalCover` for constructing of allowed Eulerian cover by trails with ordered enclosing [2].

**Algorithm OptimalCover**

**Input data:** planar graph  $G$ ;

**Output data:**  $C_j, j=1, \dots, V_{odd} / 2$  – Eulerian

cover of graph  $G$  by trails with ordered enclosing.

**Step 0.** Find the shortest matching  $M$  on the set  $V_{odd}$ .

**Step 1.** Run procedures *Initialization* and *Ordering* (i.e. Step 1 and Step 2 of algorithm `OderedEnclosing-Cover` [5]). Let  $j=1$ . Let a vertex  $v_0$  adjacent to exterior face be the current one, let  $v_0^j = v_0$ .

**Step 2.** Use procedure *Forming* [5] for constructing a trail  $C_j$  with vertex  $v_0$  as initial. Let vertex  $v_0$  be the end-vertex of constructed trail. If  $v^0 \notin V_{odd}$  then go to Step 6, otherwise go to step 3.

**Step 3.** If vertex  $v^0$  is a dead-end then go to step 5 otherwise go to step 4.

**Step 4.** If  $kmark(v_1) < kmark(v^0)$  then let  $v_0 = v^0$  and go to step 2 (continue constructing trail  $C_j$  from a current vertex).

**Step 5.** Define  $v_1: (v^0, v_1) \in M$ . Finish constructing the current trail:  $v_1^j = v^0, j = j+1, V_{odd} = V_{odd} \setminus \{v^0, v_1\}, M = M \setminus \{(v^0, v_1)\}$  and let  $v_0^j = v_1$  be a current vertex for the next trail. Go to step 2 (begin constructing a new trail  $C_j$  from vertex  $v_0 = v_0^j$ ).

**Step 6.** Stop.

**End of algorithm**

The computing complexity of `OptimalCover` algorithm does not exceed a value  $O(|V|^3)$  (i. e. computing complexity of matching constructing).

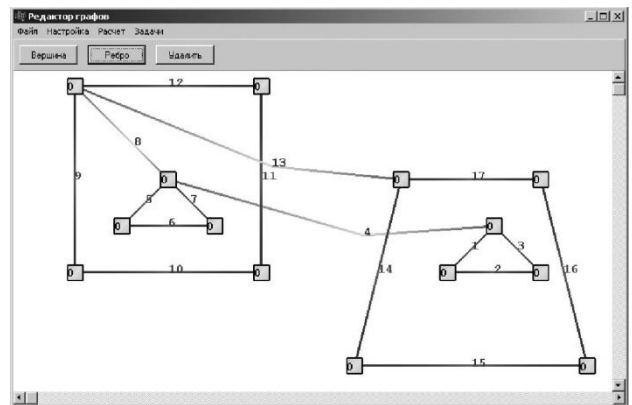
**2.2. Disconnected graph**

To solve a problem of optimization the additional edges for a disconnected graph (graph with several connected components) we need sorting of connected components according to nesting values of their edges.

**Definition 5.** As a nesting value for a connected component we shall designate the minimal nesting value of this component's edges.

After sorting the connected components according to their nesting values we get the allowed solution of a problem (see Fig. 2) by algorithm **Multi-Component-Cover** [5].

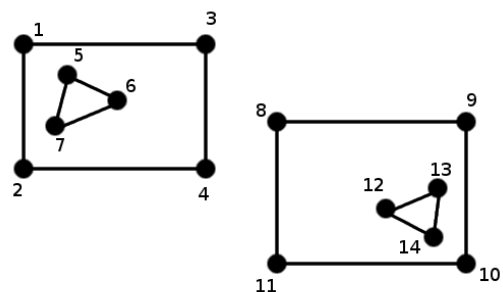
To optimize the length of additional constructions for connecting the components graph can be divided into nesting unions.



**Fig. 2.** The allowed solution for multiconnected graph

**Definition 6.** Let nesting union  $n$  be a family of connected components  $S_n$  of planar graph where connected component with nesting value  $k$  contains (encloses) connected components of nesting value greater than  $k$ .

For example, Fig. 3 represents four connected components forming two nesting unions.



**Fig. 3.** The example of nesting unions

Modifying algorithm `MultiComponentCover` according to remarks made the solution can be real-

ized as NP-hard recursive algorithm for passing all connected components. However for a problem of optimal Eulerian cover with ordered enclosing constructing it's possible to present the non-recursive algorithm solving this problem for polynomial time [6, 7].

### Algorithm OptimalMultiComponent

**Input data:** planar graph  $G$ .

**Output data:**  $C_j^s$ ,  $j=1, \dots, |V_{\text{odd}}|/2$ , – covering of graph  $G$  by trails with ordered enclosing,  $s=1, 2, \dots$  be a number of connected component.

**Step 1.** Define a set  $S$  of all connected components of graph  $G$ .

**Step 2.** Construct full abstract graph  $\mathfrak{A}$ . The connected components  $S$  of graph  $G$  be its vertices, and length of its edges is defined as a distance between the nearest vertices of these components of connectivity.

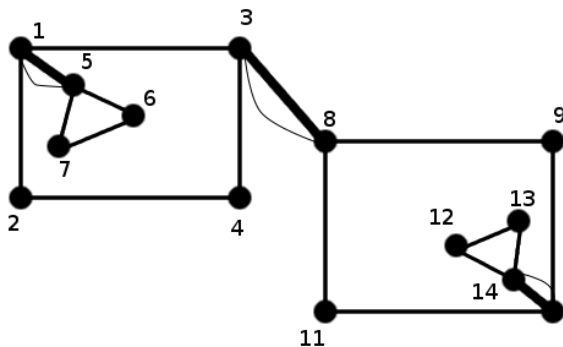
**Step 3.** Find a minimal spanning tree (MST)  $T(\mathfrak{A})$  for graph  $\mathfrak{A}$ .

**Step 4.** Add the edges of the found spanning tree to graph  $G$ :  $G_3 = G \cup T(\mathfrak{A})$ .

For example, for graph on fig.3 edges {1;5}, {3;8}, and {10;14} are added (Fig. 4, bold lines).

**Step 5.** Run algorithm OptimalCover for graph  $G_3$  (for example, Fig. 4 demonstrates the additional edges made by OptimalCover algorithm for considered graph (Fig. 3), these edges are marked by single lines).

**End of OptimalMultiComponent algorithm**



**Fig. 4.** Adding the edges connecting the components and edges to get Eulerian graph

For a graph demonstrated as an example such a cover is

$$C_1^1 = \{5, 6, 7, 5\}, C_1^2 = \{1, 3\}, C_1^3 = \{8, 9, 10\}, C_1^4 = \{14, 13, 12, 14\}, \\ C_2^3 = \{10, 11, 8\}, C_2^2 = \{3, 4, 2\}.$$

Here the upper index means the number of component, the lower index means the number of trail for this component.

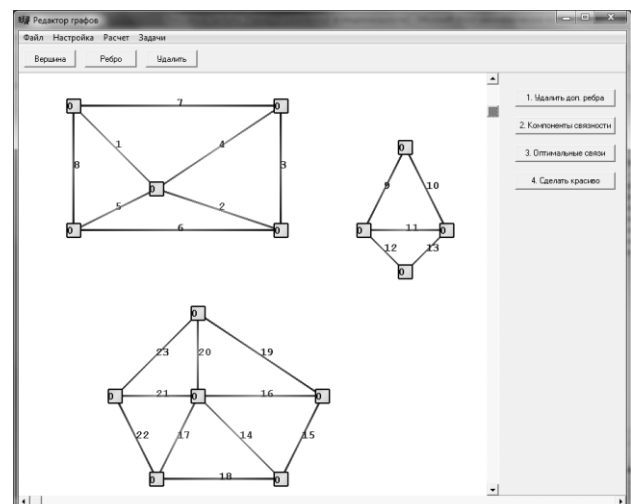
**Theorem 3.** Algorithm **OptimalMultiComponent** constructs a cover by trails with ordered enclosing by polynomial time.

**Proof.** Step 1 of algorithm represents a wave algorithm solving its task for a time  $O(|V|^2)$ . The result of Step 2 can also be gotten for the time  $O(|V|^2)$ . It's known that the solution of MST-problem (Step 3 of algorithm) can also be gotten by time  $O(|V|^2)$  or using logarithmical sorting time can be reduced to  $O(|V| \log_2 |V|)$ . Step 4 has a complexity equal to  $O(1)$ . Earlier it was mentioned that computing complexity of OptimalCover algorithm (Step 5) is not greater than  $O(|V|^3)$ . Therefore summary complexity of **OptimalMultiComponent** algorithm is  $O(|V|^3)$ .

Algorithm **OptimalMultiComponent** allows to get not worse length of additional edges than algorithm **MultiComponentCover**. It can be explained because the transition is made to the nearest allowed component instead of nearest component of the same nesting value [5].

### 3. SOFTWARE

The software named “Graph Editor” presents the primitive graphical editor for drawing plane graphs (fig.5).



**Fig. 5.** Graph Editor

This program is designed only for solving the problem of optimal cover. Pressing the corresponding buttons at right user may:

- Delete the additional edges;
- Mark the components of connectivity;

Optimize the additional constructions between the components (Kruskal's algorithm);

Demonstrate all the additional constructions.

Using the main menu a user may solve a problem using optimization or not. In the second case one may receive the allowed solution (Fig. 6). All the connected components are defined, the MST between the components is constructed. All the edges are marked by numbers of their order in a resulting cover.

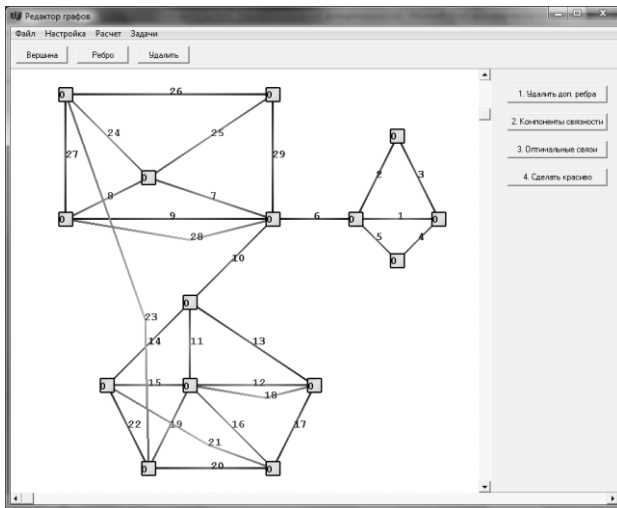


Fig. 6. Running of algorithm

User may also animate the answer to show the results as a presentation.

#### 4. CONCLUSION

So the algorithms for constructing the covers with ordered enclosing for different types of plane graphs are reviewed. It is shown that all of these algorithms allow effectively solve the problems of cutting for any cutting plan. The developed software demonstrates the work of these algorithms.

#### ACKNOWLEDGMENTS

The research is held under support of Ministry of Education and Science of Russian Federation, contract No. 14.V37.21.0395.

#### REFERENCES

1. T. Panyukova, "Cover with ordered enclosing for flat graphs," *iElectronic Notes in Discrete Mathematics*, no. 28, pp. 17-24, 2007.
2. T. A. Panyukova, "Software for constructing of chains with ordered enclosing," (in Russian), in *Proc. of 39-th Regional Conference on Problems of Theoretical and Applied Mathematics*, IMM, Ural Branch of RAS, Yekaterinburg, 2008, pp. 360-365.
3. T. A. Panyukova, "Optimization of resources on technological support of cutting process," (in Russian), *Applied Informatics*, no. 3 (39), pp. 20-32, 2012.
4. T. A. Panyukova, "Optimal eulerian coverings for planar graphs," (in Russian), *Discrete Analysis and Operation Research*, vol. 18, no. 2, pp. 64-74, 2011.
5. T. A. Panyukova and E. A. Savitskiy, "Allowed eulerian coverings with ordered enclosing for a multiconnected graph," (in Russian), in *Proc. of Russian Conference on Statistics, Modeling, and Optimization (Chelyabinsk, November, 28 – December, 2, 2011)*, Chelyabinsk: Publishing Center of SUSU, 2011, pp. 154-159.
6. T. A. Panyukova, "The algorithm for constructing the optimal eulerian cover for multiconnected graph," (in Russian), in *Papers of VII Int. Conf. "Modern Information Technologies and IT-education"*, Moscow: "INTUIT.ru", 2012.
7. T. A. Panyukova, "About optimal eulerian cover of plane graphs," (in Russian), in *Papers of Int. Scientific Conf. «Information Technologies and Systems»*: Scientific electronic issue (64 files), Chelyabinsk: CSU-Publishing, 2013, pp. 52-54.

#### ABOUT AUTHORS

**PANYUKOVA, Tatiana Anatolievna**, Ass.Prof., Mathematical Methods in Economics and Statistics (Chelyabinsk, South Ural State University, 2003). Cand. of Phys. & Math. Sci. (Moscow, Computing Center of RAS, 2006).

**SAVITSKIY, Egor Aleksandrovich**, Postgrad. (PhD) Student, Dept. of Mathematical Methods in Economics and Statistics. Economist-mathematician (Chelyabinsk, South Ural State University, 2013).