On Triangle Cover Contact Graphs

Shaheena Sultana[1⋆], Md. Iqbal Hossain[2], Md. Saidur Rahman[1], Nazmun Nessa Moon[1] and Tahsina Hashem[1]

[1] Graph Drawing and Information Visualization Laboratory,
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology (BUET),
Dhaka-1000, Bangladesh
zareefas.sultana@gmail.com, saidurrahman@cse.buet.ac.bd,
moon_ruet@yahoo.com, tahsinahashem@gmail.com
[2] The University of Arizona, USA
hossain@email.arizona.edu

Abstract. Let $S = \{p_1, p_2, \ldots, p_n\}$ be a set of pairwise disjoint geometric objects of some type in a $2D$ plane and let $C = \{c_1, c_2, \ldots, c_n\}$ be a set of closed objects of some type in the same plane with the property that each element in C covers exactly one element in S and any two elements in C are interior-disjoint. We call an element in S a *seed* and an element in C a *cover*. A *cover contact graph (CCG)* has a vertex for each element of C and an edge between two vertices whenever the corresponding cover elements touch. It is known how to construct, for any given point seed set, a disk or triangle cover whose contact graph is 1- or 2-connected but the problem of deciding whether a k-connected CCG can be constructed or not for $k > 2$ is still unsolved. A *triangle cover contact graph (TCCG)* is a cover contact graph whose cover elements are triangles. In this paper, we give algorithms to construct a 3-connected $TCCG$ and a 4-connected $TCCG$ for a given set of point seeds. We also show that any connected outerplanar graph has a realization as a $TCCG$ on a given set of collinear point seeds. Note that, under this restriction, only trees and cycles are known to be realizable as CCG.

⋆ She was supported by ICT Fellowship, Ministry of Posts, Telecommunications and IT, Government of the People's Republic of Bangladesh.

1 Introduction

Let $S = \{p_1, p_2, \ldots, p_n\}$ be a set of pairwise disjoint geometric objects of some type in the plane and let $C = \{c_1, c_2, \ldots, c_n\}$ be a set of closed objects of some type in the same plane with the property that each element in C covers exactly one element in S and any two elements in C can intersect only on their boundaries. We call an element in S a *seed* and an element in C a *cover*. The seeds may be points, disks or triangles and covering elements may be disks or triangles. The *cover contact graph (CCG)* consists of a set of vertices and a set of edges where each vertex corresponds to a cover and each edge corresponds to a connection between two covers if they touch at their boundaries. In other words, two vertices of a cover contact graph are adjacent if the corresponding cover elements touch at their boundaries. Note that the vertices of the cover contact graph are in one-to-one correspondence to both seeds and covering objects. In a cover contact graph, if disks are used as covers then it is called a *disk cover contact graph* and if triangles are used as covers then it is called a *triangle cover contact graph (TCCG)*. Figure 1(b) depicts the disk cover contact graph induced by the disk covers in Fig. 1(a), whereas Fig. 1(d) depicts the triangle cover contact graph induced by the triangle covers in Fig. 1(c). A *coin graph* is a graph formed by a set of disks, no two of which have overlapping interiors, by making a vertex for each circle and an edge for each pair of circles that touches. Koebe's theorem [7, 9] states that every planar graph can be represented as a coin graph. There are several works [10, 11, 5] in the geometric-optimization community where the problem is how to cover geometric objects such as points by other geometric objects such as convex shapes, disks. The main goal is to minimize the radius of a set of k disks to cover n input points. Applications

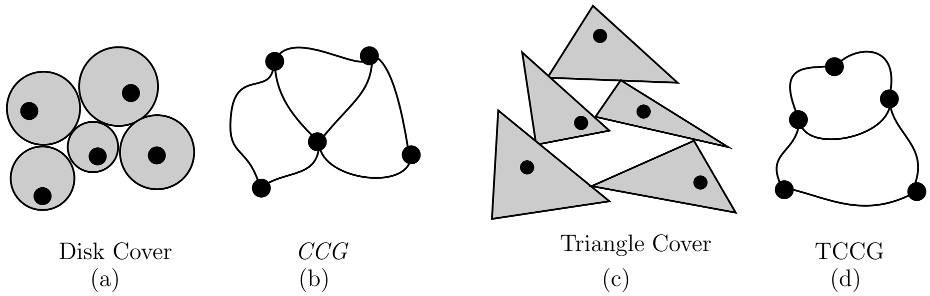

Fig. 1. Illustration for *CCG* and *TCCG*; (a) a disk cover, (b) a CCG, (c) a triangle cover and (d) a TCCG.

of such covering problems are found in geometric optimization problems such as facility location problems [10, 11]. Abellanas *et al.* [1] worked on a "coin placement problem," which is NP-complete. They tried to cover n points using n disks (each having different radius) by placing each disk in the center position at one of the points so that no two disks overlap. Further Abellanas *et al.* [2] considered another related problem. They showed that for a given set of points in the plane, it is also NP-complete to decide whether there are disjoint disks centered at the points such that the contact graph of the disks is connected.

Recently, Atienza *et al.* [3] introduced the concept of cover contact graphs where the seeds are not necessarily the center of the disks. They gave an $O(n \log n)$ time algorithm to decide whether a given set of point seeds can be covered with homothetic triangles or disks such that the resulting cover contact graph is 1- or 2-connected. The k-connectivity problem is still unsolved for $k > 2$. Atienza *et al.* [3] also considered the problem from another direction which they called "realization problem." In a realization problem we are given a graph G of n vertices and a set S of n seeds and we are asked whether there is any covering so that the resulting cover contact graph is G. They gave some necessary conditions and then showed that it is NP-hard to decide whether a given graph can be realized as a disk cover contact graph if the correspondence between vertices and point seeds is given. They also showed that every tree and cycle have realizations as CCGs on a given set of collinear point seeds. Durocher *et al.* [5] considered a circular cover contact graph problem defined by Atienza *et al.* [3]. They showed that when the input discs and the covering discs are all constrained to touch a line, then the problem of deciding whether the input set has a connected CCG is NP-hard. They also defined an approximate variation of the problem, where the covering discs are allowed to overlap by a small amount. They gave a polynomial-time algorithm such that if there exists an exact solution to the problem, then the algorithm returns an ϵ-approximate solution.

In this paper, we consider a set of arbitrary seeds in the plane where the seeds are points and the covers are triangles. First we consider the set of seeds which are in general position, i.e, no two seeds lie on a vertical line and we give an $O(n \log n)$ algorithm to construct a 3-connected $TCCG$ of the set of seeds. We also give a $O(n \log n)$ algorithm to construct a 4-connected $TCCG$ for a given set of six or more seeds. Addressing the realization problem, we give an algorithm that realizes a given outerplanar graph as a triangle cover contact graph ($TCCG$) for a given set of seeds on a line.

The remaining of the paper is organized as follows. Section 2 presents some definitions and preliminary results. Section 3 gives algorithms to construct a 3-connected $TCCG$ and 4-connected $TCCG$. Section 4 gives an algorithm that realizes a given outerplanar graph as $TCCG$. Finally, Section 5 concludes the paper by suggesting some future works. A preliminary version of this paper was presented at WALCOM 2015 [6].

2 Preliminaries

In this section we present some terminologies and definitions which will be used throughout the paper. For the graph theoretic definitions which have not been described here, see [4, 8].

A graph is *planar* if it can be embedded in the plane without edge crossing except at the vertices where the edges are incident. A *plane graph* is a planar graph with a fixed planar embedding. A plane graph divides the plane into connected regions called *faces*. The unbounded region is called the *outer face*; the other faces are called *inner faces*. The cycle lies on the outer face is called *outer cycle*. A plane graph G is an *outerplanar graph* if all vertices of G lie on the outer face.

The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single-vertex graph. We say that G is k-connected if $\kappa(G) \geq k$. A vertex v in a connected graph G is a *cut-vertex* if the deletion of v from G results in a disconnected graph. Similarly an edge e in a connected graph G is a *bridge* if the deletion of e from G results in a disconnected graph. A 2-connected or biconnected graph does not contain any cut vertex.

A *biconnected component* of a connected graph G is a maximal biconnected subgraph of G. A *block* of a connected graph G is either a biconnected component or a bridge of G. The graph in Fig. 2(a) has the blocks $B_0, B_1, \ldots, B_8$ depicted in Fig. 2(b). The blocks and cut vertices in G can be represented by a tree T, called the *BC-tree* of G. In T each block is represented by a B-node and each cut vertex of G

is represented by a C-node. The BC-tree T of the plane graph G in Fig. 2(a) is depicted in Fig. 2(c), where each B-node is represented by a rectangle and each C-node is represented by a circle.

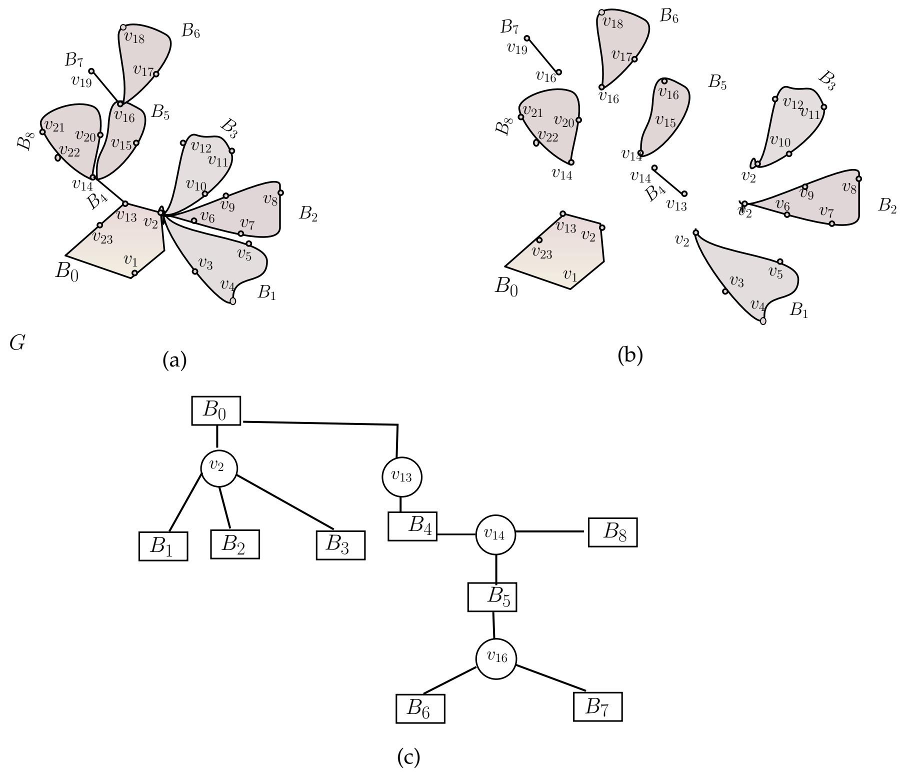

Fig. 2. (a) A connected graph G, (b) blocks of G and (c) BC-tree T.

Let $S = \{p_1, p_2, \ldots, p_n\}$ be a set of pairwise disjoint geometric objects of some type and $C = \{c_1, c_2, \ldots, c_n\}$ be a set of closed objects of some type with the property that each element in C covers exactly one element in S and any two elements in C can intersect only on their boundaries. We call an element in S a *seed* and an element in C a *cover*. The seeds may be points, disks or triangles and covering elements may be disks or triangles. The *cover contact graph (CCG)* induced by C is the contact graph of the elements of C, that is, the graph $G = (C, E)$ with $E = \{\{C_i, C_j\} \subseteq C \mid C_i \neq C_j, C_i \cap C_j \neq \emptyset\}$. In other words, two vertices of a cover contact graph are adjacent if the corresponding cover elements touch at their boundaries.

Note that the vertices of the cover contact graph are in one-to-one correspondence to both seeds and covering objects. In a cover contact graph, if disks are used as covers then it is called a *disk cover contact graph* and if triangles are used as covers then it is called a *triangle cover contact graph (TCCG)*. Figure 1(c) shows a triangle cover of seeds and Fig. 1(d) shows the resulting $TCCG$.

For a point a in a $2D$ plane, we denote by x_a and y_a the x-coordinate and the y-coordinate of a, respectively. We specify the position of a in the plane by (x_a, y_a). We denote the straight-line segment that passes through the points a and b by L_{ab}. We denote a triangle connecting three points a_1, a_2, a_3 by $T(a_1, a_2, a_3)$ and a trapezoid connecting four points a_1, a_2, a_3, a_4 by $trap(a_1, a_2, a_3, a_4)$. We denote a path in a simple graph by the ordered sequence of vertices on the path.

We work on connectivity problem of a given set of seeds. Here we use the set of seeds which are in general position, i.e., no two seeds lie on a vertical line. We first show that a set of seeds admits a path $TCCG$. First, we sort the seeds according to x-coordinate value. Then we cover the seeds with triangles such that every triangle touches the previous and next triangle (except the first and the last triangle). The contact graph of the triangles forms a path. We have the following trivial lemma.

Lemma 1. *Let S be a set of seeds in general position where no two seeds are on a vertical line. Then S admits a path $TCCG$.*

Proof. Let $p_1, p_2, \ldots, p_n$ be the seeds of S according to their left-to-right order. We can cover seed p_i by triangle T_i as follows. We draw the vertical segment of T_i through p_i and then add the third point of T_i on vertical segment of T_{i-1} to complete T_i. This creates n triangles covering S. A triangle touches its previous triangle as shown in Fig. 3. Hence the resultant graph is a path. □

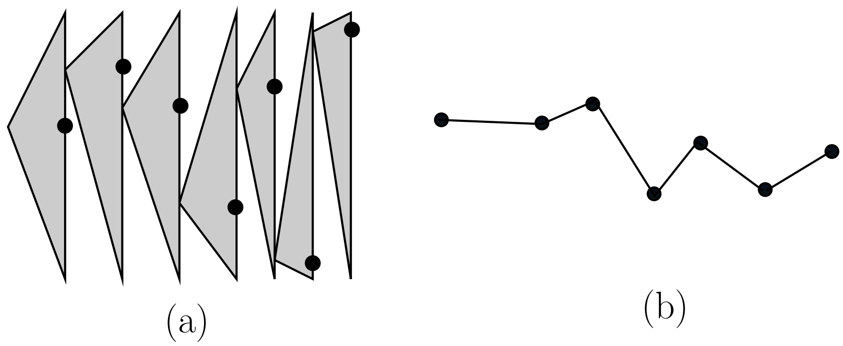

Fig. 3. (a) S is covered by triangles and (b) a path $TCCG$.

3 k-connected $TCCG$

In this section we work on the connectivity problem of a given set of seeds. Atienza *et al.* [3] have given an algorithm to decide whether a given set of point seeds can be covered with homothetic triangles or disks such that the resulting cover contact graph is 1- or 2-connected. In Section 3.1 and Section 3.2 we develop $O(n \log n)$ time algorithms to show that a set of given seeds can always be covered with triangles such that the resulting cover contact graphs are 3-connected and 4-connected, respectively.

3.1 3-connected *TCCG*

In this section we are given a set S of seeds in general position and we have to decide whether S admits a 3-connected $TCCG$. Before presenting the details of our algorithm we give an outline of the algorithm. First we sort the seeds $p_1, p_2, \ldots, p_n$ according to their left-to-right order. Then we cover p_1, p_{n-1} and p_n points by three mutually touching triangles T_1, T_{n-1} and T_n such that all other points are in the closed area as illustrated in Fig. 4(b). Then each point of S inside the closed area is covered by a triangle such that it touches three triangles. We now give the following theorem whose proof immediately gives a formal algorithm.

Theorem 1. *Let S be a set of seeds where no two seeds are on a vertical line then S admits a 3-connected $TCCG$. Furthermore, it can be found in $O(n \log(n))$ time.*

Proof. We first build a right triangle $\triangle$ containing $n-3$ seeds from S such that each outside of $\triangle$ contains one seed as shown in Fig. 4(b). This can be always done by rotating S, if necessary. We cover each seed inside of $\triangle$ by a triangle in the same way as described in Lemma 1. Additionally, we can change two points of each triangle such that they touch hypotenuse and adjacent side of $\triangle$. Then outer three seeds of $\triangle$ can be covered by three triangles as shown in Fig. 4(c). The resultant graph of the seeds in Fig. 4(a) is shown in Fig. 4(d). Since each triangle inside of $\triangle$ touches two outside triangles and the previous triangle, the resultant graph is a 3-connected graph. For sorting we need $O(n \log(n))$ time.

$\square$

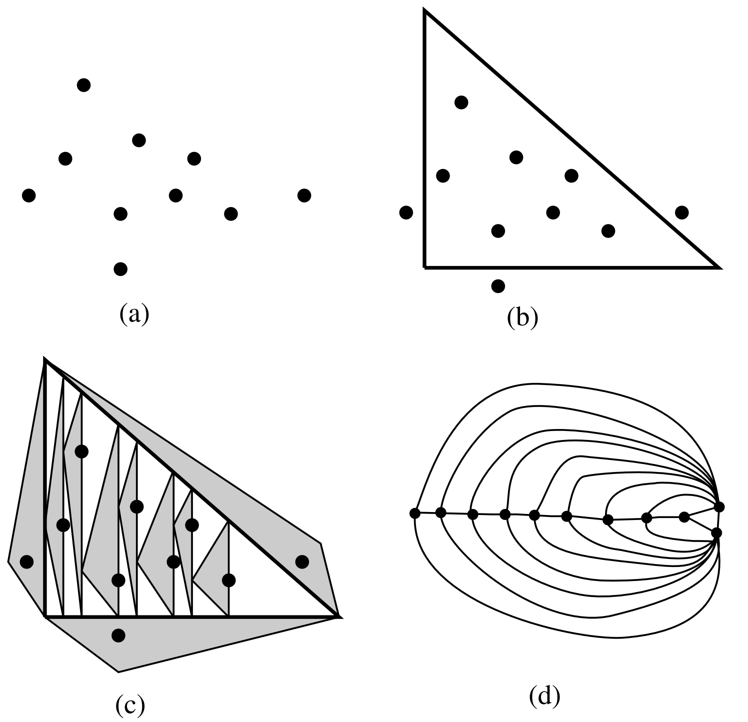

Fig. 4. (a) A set of seeds S, (b) a right angle containing $n-3$ seeds, (c) S is covered by triangles and (d) 3-connected $TCCG$.

3.2 4-connected *TCCG*

In this section we show that every set S of seeds such that no two seeds are on a vertical line admits a 4-connected $TCCG$.

Theorem 2. *Let S be a set of six or more seeds such that no two seeds are on a vertical line nor on a horizontal line. Then S admits a 4-connected $TCCG$, and such a $TCCG$ can be found in $O(n \log n)$ time.*

Proof. We use similar technique as described in Theorem 1. In this case we take a trapezoid ⏢ rather than a triangle such that one seed lies each outside of ⏢ as shown in Fig. 5(a), and rest $n-4$ seeds lie inside of ⏢. In this trapezoid we can ensure that the left and the right side are not parallel to each other. Now for covering seeds inside of ⏢, we follow Lemma 1 and modify triangles such that two points of each triangle touch two horizontal segments of ⏢ as shown in Fig. 5(b). The four outside seeds of ⏢ can be covered by four triangles such that the left and the right side triangle touch each other (at the top or down side of ⏢). This can be done because left and right side of ⏢ are not parallel.

We now label the vertices of the resultant graph as follows. We label v_t and v_b for the corresponding top and bottom triangles of ⏢, respectively. Then rest of the vertices are labeled as $v_1, v_2, \ldots, v_{n-2}$ for the corresponding triangles using left to right order.

Since S has six or more vertices, inside of ⏢ has at least two vertices and hence every vertex of G has degree at least four. Thus to prove our claim that G is 4-connected, it is now sufficient to show that four internally vertex-disjoint paths exist between each of the pairs $\{v_i, v_j\}$, $\{v_i, v_t\}$, $\{v_i, v_b\}$ and $\{v_t, v_b\}$ of vertices, where $1 \leq i < j \leq k$, (see Fig. 5(d)). The paths between the vertices v_i

and v_j are path $(v_i, v_{i+1}, \ldots, v_j)$, path (v_i, v_t, v_j), path (v_i, v_b, v_j) and path $(v_i, v_{i-1}\ , \ldots, v_1, v_k, v_{k-1}\ , \ldots, v_j)$. The paths between the vertices v_1 and v_t are path (v_1, v_t), path (v_1, v_2, v_t), path (v_1, v_k, v_t) and path (v_1, v_b, v_3, v_t). The paths between the vertices v_k and v_t are path (v_k, v_t), path (v_k, v_{k-1}, v_t), path (v_k, v_1, v_t) and path (v_k, v_b, v_{k-2}, v_t). The paths between the vertices v_i and v_t are path (v_i, v_t), path (v_i, v_{i-1}, v_t), path (v_i, v_{i+1}, v_t) and path (v_i, v_b, v_k, v_t) if $i \neq k-1$ or path (v_i, v_b, v_1, v_t) if $i \neq 2$. Similarly there are at least four internally vertex-disjoint paths exist between vertices v_i and v_b. It is not difficult to see that k paths exist between v_t and v_b through v_i. Thus four internally vertex-disjoint paths exist between each pair of vertices.

□

Our construction for a 4-connected $TCCG$ given in the proof of Theorem 2 can be used for any set of six or more seeds; if two seeds are on a vertical line or on a horizontal line then rotate the plane such that no two points remain on a vertical line or on a horizontal line.

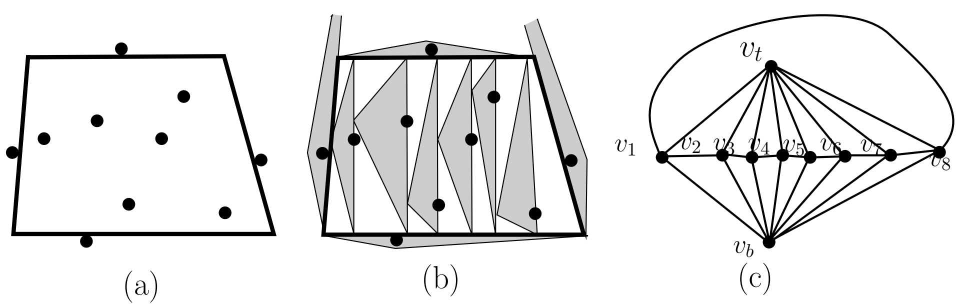

Fig. 5. (a) A set of seeds S and a trapezoid which encloses $n-4$ seeds, (b) S is covered by triangles and (c) 4-connected $TCCG$.

4 Realizability of Outerplanar Graphs

In this section we show that a connected outerplanar graph has a realization as a triangle cover contact graph (TCCG) on a given set of seeds on a line as in Theorem 4. To prove this theorem we need the following Theorem.

Theorem 3. *Let G be a biconnected outerplanar graph of n vertices. Let S be a set of n seeds on a horizontal line. Then G is realizable on S as a TCCG in $O(n \log n)$ time.*

Proof. We give a constructive proof. Let $v_1, v_2, \ldots, v_n$ be the vertices on the outer face of G in anti-clockwise order. The starting vertex can be chosen arbitrarily. Let $p_1, p_2, \ldots, p_n$ be the seeds of S sorted according to their x-coordinates.

We cover each seed p_i by a covering triangle $T_i(a_i, b_i, c_i)$ corresponding to the vertex v_i such that a_i is drawn on p_i, the side $L_{a_i b_i}$ of T_i is vertical, $x_{c_i} < x_{a_i}, x_{b_i}$ and $y_{a_i} < y_{b_i}, y_{c_i}$. That is, the right side $L_{a_i b_i}$ of the triangle T_i is vertical and c_i lies in the left half-plane of $L_{a_i b_i}$. We perform the task in two steps: realizing the outer cycle, and realizing inner edges of G.

Realizing outer cycle: We draw T_1 and T_n by covering p_1 and p_n such that the point c_n of T_n meet on the $L_{a_1 b_1}$ of T_1. For the rest of the seeds we follow similar technique describe in Lemma 1. That gives realization of the path $v_2, v_3, \ldots, v_{n-1}$. Now we change T_2 and T_{n-1} such that they touch T_1 and T_n, respectively (see Fig. 6(c)).

Realizing inner edges: If v_i, v_j is an inner edge then we modify b_i of T_i and c_j of T_j to touch each other without loosing connectivity of other triangles (see Fig. 6(d)). This operation is always possible because any edge (v_p, v_q) does not exist for $n \geq p > j$ and $j > q > i$, and for $1 \leq p < i$ and $j > q > i$. Note that G is a planar graph.

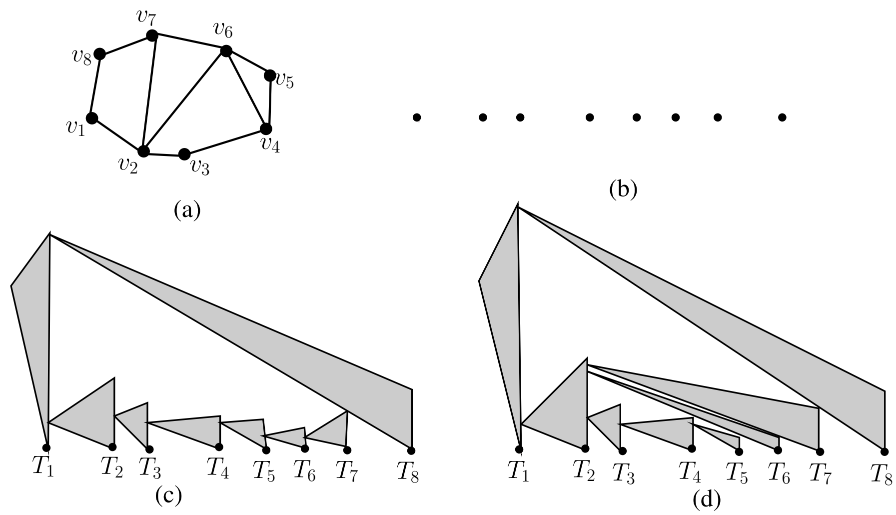

Fig. 6. a) A biconnected outerplanar graph G, (b) a set of seeds S, (c) realization of the outer cycle of G as $TCCG$ on S and (d) realization of G as $TCCG$ on S.

Note that in a realization of a biconnected outerplanar graph G by the algorithm described above, the triangle T_1 corresponding to the start vertex and the triangle T_n corresponding to the end vertex touch each other on vertical segment of the triangle T_1. For inner edges, the triangles are adjusted without overlapping. One can observe that the algorithm works when the vertical side of the triangle $T_1(a_1, b_1, c_1)$ covering the first seed p_1 is already fixed. In that case the point b_1 of T_1 might be moved upward to make the contact with the triangle T_n.

All this can be done in $O(n)$ time, but for sorting we need $O(n \log n)$ time.

$\square$

Using Theorem 3 we prove the following theorem.

Theorem 4. *Let G be a connected outerplanar graph of n vertices. Let S be a set of n collinear seeds. Then G is realizable on S as a TCCG in $O(n \log n)$ time.*

Proof. We give a constructive proof as follows. Let G be a connected outerplanar graph. We first decompose G into blocks and construct a BC-tree. We make the BC-tree a rooted BC-tree by making an arbitrary B node as the root. Let B_0 be the root of the BC-tree. A rooted BC-tree with the root B_0 for the graph in Fig. 7(a) is illustrated in Fig. 7(b). We next traverse the BC-tree by DFS and get an ordering of blocks and cut vertices. A DFS ordering of the blocks and cut vertices of the BC-tree in Fig. 7(b) is $B_0, v_2, B_1, B_2, B_3, v_{13}, B_4, v_{14}, B_5, v_{16}, B_6, B_7, B_8$. We next traverse the vertices on the outer face of G in anti-clockwise order starting from any vertex of B_0. Let $v_1, v_2, \ldots, v_n$ be the vertices on the outer face of G in anti-clockwise order where only the first appearance of a cut vertex is kept in the ordering, as illustrated in Fig. 7(a). Note that a cut vertex may appear several times when we traverse the outer face, but only the first appearance is kept in the ordering. We assume that the seeds in S are on a horizontal line (otherwise, we rotate the plane such that the seeds in S are on a horizontal line). Let $p_1, p_2, \ldots, p_n$ be the seeds of S sorted according to their x-coordinates. We assign seeds p_i to vertices v_i for $1 \leq i \leq n$ as illustrated in Fig. 7(c). We next realize each biconnected component on its assigned seeds by the algorithm described in Theorem 3. Each bridge contains only two vertices and we realize it trivially. Note that the blocks are realized in their DFS order. After realizing B_0, the triangle corresponding to a cut vertex is realized at the left of its children blocks in the BC-tree. While realizing B_i by the algorithm described in the proof of the Theorem 3, the parent cut vertex, which is already realized as a triangle, is taken as the start vertex of B_i. In Fig. 7(c), when we realize B_0, the triangle corresponding to cut vertices v_2 and v_{13} of G is realized. While realizing B_1, B_2 and B_3 by the algorithm described in the proof of the Theorem 3, the cut vertex v_2, which is already realized as a triangle, is taken as the start vertex of B_1, B_2 and B_3. The triangle corresponding to the end vertices of B_1, B_2 and B_3 touch the vertical segment of the triangle corresponding to the start vertex v_2.

One can observe that the seeds assigned to the vertices (except the start vertex) of a leaf block are consecutive and the seeds assigned to the vertices of a block remain inside the span of the seeds assigned to the vertices of its ancestor blocks. Thus a block is realized inside the realization of its ancestor blocks. Therefore Theorem 3 and the ordering of the blocks ensure the correctness of the algorithm.

We now analyze the time complexity of our construction. For computing the triangle covers of n seeds we need $O(n)$ time. Since we are sorting the seeds, the realization can be found in $O(n \log n)$ time.

□

5 Conclusion

In this paper we have developed $O(n \log n)$ time algorithms to show that a set of given seeds can always be covered with triangles such that the resulting cover contact graph is 3-connected. We also have shown that any set of six or more point seeds admits covers with 4-connected $TCCG$ and such covers can be found in $O(n \log n)$ time. We also have shown that every connected outerplanar graph can be realized as a $TCCG$ of a set of point seeds on a straight line. Following interesting open problems have come out from this work.

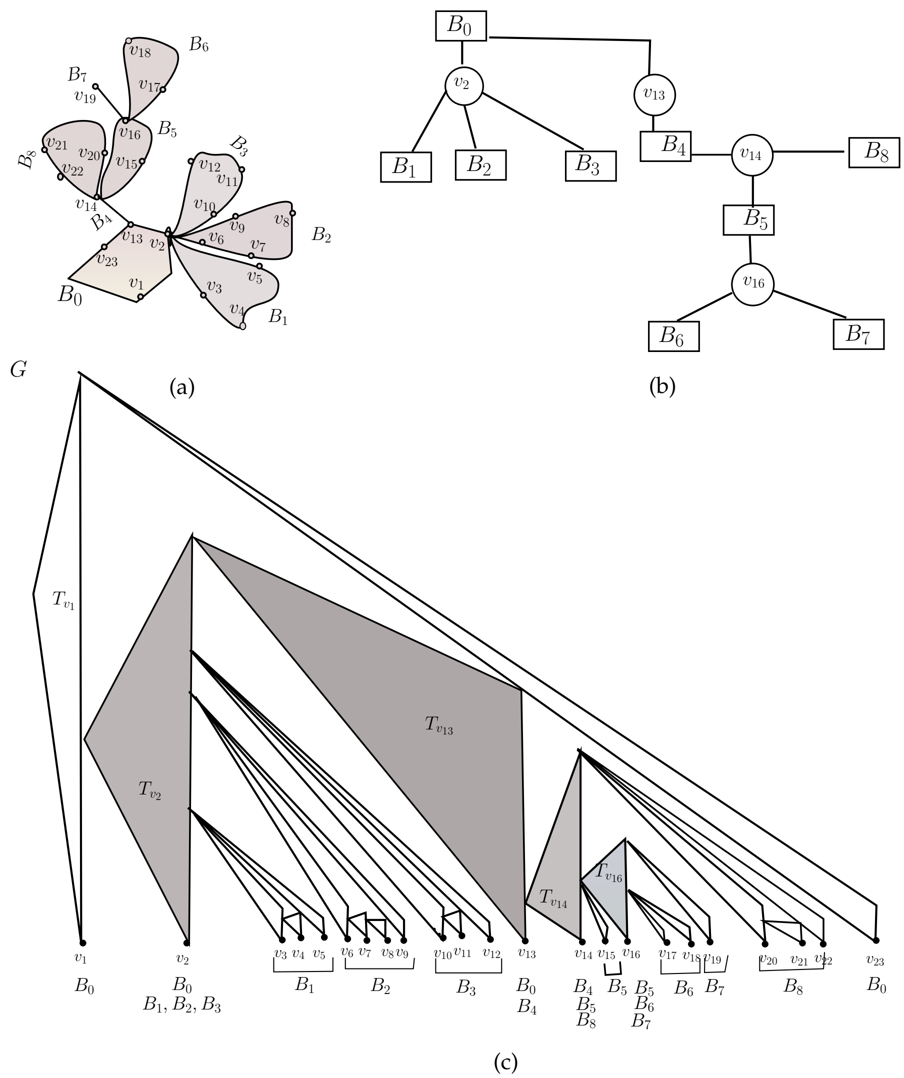

Fig. 7. a) A graph with biconnected components, b) BC-tree of the graph and c) realiziation of the graph as $TCCG$ on S.

1. Can we give lower bounds on the smallest angles needed?
2. Can some of the results be achieved with more restricted triangle classes like homothetic triangles?
3. We can investigate which larger classes of graphs are realizable as $TCCG$.
4. What about other classes of seeds and covers? Is realization is possible if seeds are in general position?

Acknowledgement

We thank anonymous referees for their valuable comments for ensuring the correctness and improving the presentation of the paper.

References

1. M. Abellanas, S. Bereg, F. Hurtado, A. G. Olaverri, D. Rappaport, and J. Tejel. Moving coins. *Comput. Geom.*, 34(1):35–48, 2006.
2. M. Abellanas, N. Castro, G. Hernández, A. Márquez, and C. Moreno-Jiménez. *Gear System Graphs.* Manuscript. 2006.
3. N. Atienza, N. Castro, C. Cortés, M. A. Garrido, C. I. Grima, G. Hernández, A. Márquez, A. Moreno, M. Nöllenburg, J. R. Portillo, P. Reyes, J. Valenzuela, M. Trinidad Villar, and A. Wolff. Cover contact graphs. *Journal of Computational Geometry*, 3(1):102–131, 2012.
4. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall Inc., 1999.
5. S. Durocher, S. Mehrabi, M. Skala, and M. A. Wahid. The cover contact graph of discs touching a line. In *CCCG*, pages 59–64, 2012.
6. M. I. Hossain, S. Sultana, N. N. Moon, T. Hashem, and M. S. Rahman. On triangle cover contact graphs. In *Algorithms and Computation - 9th International Workshop, WALCOM 2015*, volume 8973 of *LNCS*, pages 323–328. Springer, 2015.
7. P. Koebe. Kontaktprobleme der konformen abbildung. *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Klasse*, 88(1-3):141–164, 1936.
8. T. Nishizeki and M. S. Rahman. *Planar Graph Drawing.* Lecture Notes Series on Computing. World Scientific, Singapore, 2004.
9. J. Pach and P. K. Agarwal. *Combinatorial Geometry.* John Wiley and Sons, New York, 1995.

10. J. M. Robert and G. T. Toussaint, editors. *Computational geometry and facility location*, volume 68. In Proc. Int. Conf. Oper. Res. Manage. Sci., 1990.
11. E. Welzl. Smallest enclosing disks (balls and ellipsoids). In I. H. Maurer, editor, *New Results and New Trends in Computer Science*, Lect. Notes Comput. Sci., pages 359–370. Springer-Verlag, 1991.