On Touching Triangle Graphs

Emden R. Gansner[1], Yifan Hu[1], and Stephen G. Kobourov[2]

[1] AT&T Labs - Research, Florham Park, NJ
{erg, yifanhu}@research.att.com
[2] University of Arizona, Tucson, AZ
kobourov@cs.arizona.edu

Abstract. In this paper, we consider the problem of representing graphs by triangles whose sides touch. We present linear time algorithms for creating touching triangles representations for outerplanar graphs, square grid graphs, and hexagonal grid graphs. The class of graphs with touching triangles representations is not closed under minors, making characterization difficult. We do show that pairs of vertices can only have a small common neighborhood, and we present a complete characterization of the subclass of biconnected graphs that can be represented as triangulations of some polygon.

Keywords: planar graphs, contact graphs

1 Introduction

Planar graphs are a widely studied class that includes naturally occurring subclasses such as trees and outerplanar graphs. Typically planar graphs are drawn using the node-link model, where vertices are represented by points and edges are represented by line segments. Alternative representations, such as contact circles [5] and contact triangles [8] have also been explored. In these representations, a vertex is a circle or triangle, and an edge is represented by pairwise contact at a common point.

In this paper, we explore the case where vertices are polygons, with an edge whenever the sides of two polygons touch. Specifically, given a planar graph $G = (V, E)$, we would like to find a set of polygons R such that there is bijection between V and R, and two polygons touch non-trivially if and only if the corresponding vertices are adjacent in G.

Note that, unlike the case of contact circle and contact triangle representations, two polygons that share a common point are not considered adjacent. In the sequel, we use "contact" to refer to edges touching non-trivially.

A theorem of Thomassen [20] implies that all planar graphs can be represented using convex hexagons and this also follows from results by Kant [15] and de Fraysseix *et al.* [7]. Gansner *et al.* [10] have shown that six sides are not only sufficient but also necessary, and gave a linear time construction. This leads us to consider which planar graphs can be represented by polygons with fewer than six sides.

This paper presents some initial results for the case of touching triangles. We assume we are dealing with connected planar graphs $G = (V, E)$. We let TTG denote the class of graphs that have a touching triangles representation. Concerning how to attack the

problem, we can start with some simple observations. First, unlike such classes as planar graphs, TTG graphs are not closed under homeomorphisms or minors (cf. Appendix). On the other hand, as Corollary 1 shows, we can sometimes find a subclass of TTG graphs which can be extended by homeomorphism. In addition, there is the special subclass of *filled* TTG graphs, i.e., those for which the polygon formed by the union of triangles is simply-connected. At times, the filled version can be more tractable than the general version, and may lead to a solution of the general problem [6].

In Section 2, we show that all outerplanar graphs can be represented as filled TTG. Similarly, we show in Section 3 that all subgraphs of a square or hexagonal grid are in TTG. All of these representations can be computed in linear time. Section 4 characterizes the special case of graphs arising from filled triangulations of polygons. Finally, in Section 5, we show that, for graphs in TTG, pairs of vertices can have very limited common neighborhoods. This allows us to identify concrete examples of graphs not in TTG.

1.1 Related Work

In the limiting case, one can date results on representing planar graphs as touching polygons to Koebe's 1936 theorem [16] which states that any planar graph can be represented as a contact graph of disks in the plane. Kant's linear time algorithm for drawing degree-3 planar graphs on a hexagonal grid [15] can be used to obtain hexagonal drawings for planar graphs. Gansner *et al.* [10] show that at least six sides are necessary and that the lower bound is matched by an upper bound of six sides with a linear time algorithm for representing any planar graph by touching convex hexagons.

The problem restricting the polygons to isothetic rectangles has been extensively studied, starting with Ungar [21]. Rahman *et al.* [18] describe a linear time algorithm for constructing rectangular contact graphs, if one exists. Buchsbaum *et al.* [6] provide a characterization of the class of graphs that admit rectangular contact graph representation. The version of the problem where it is further required that the rectangles partition a rectangle is known as the rectangular dual problem. Bhasker and Sahni[4] and He [12] describe linear time algorithms for constructing a rectangular dual of a planar graph, if one exists.

In VLSI floor-planning it is often required to partition a rectangle into rectilinear regions so that non-trivial region adjacencies correspond to a given planar graph. It is natural to try to minimize the complexities of the resulting regions and the best known results are due to He [13] and Liao *et al.* [17] who show that regions need not have more than 8 sides. Both of these algorithms run in $O(n)$ time and produce layouts on an integer grid of size $O(n) \times O(n)$, where n is the number of vertices.

Rectilinear cartograms can be defined as rectilinear contact graphs for vertex weighted planar graphs, where the area of a rectilinear region must be proportional to the weight of its corresponding node. Even with this extra condition, de Berg *et al.* [2] show that rectilinear cartograms with constant region complexity can be constructed in $O(n \log n)$ time. Specifically, a rectilinear cartogram with region complexity 40 can always be found.

2 Outerplanar Graphs

In this section, we show that any outerplanar graph can be represented by a set of touching triangles, that is, outerplanar graphs belong to the class TTG. Here we assume that we are given an outerplanar graph $G = (V, E)$ and the goal is to represent G as a set of touching triangles. We describe a linear time algorithm based on inserting the vertices of G is an easy-to-compute "peeling" order.

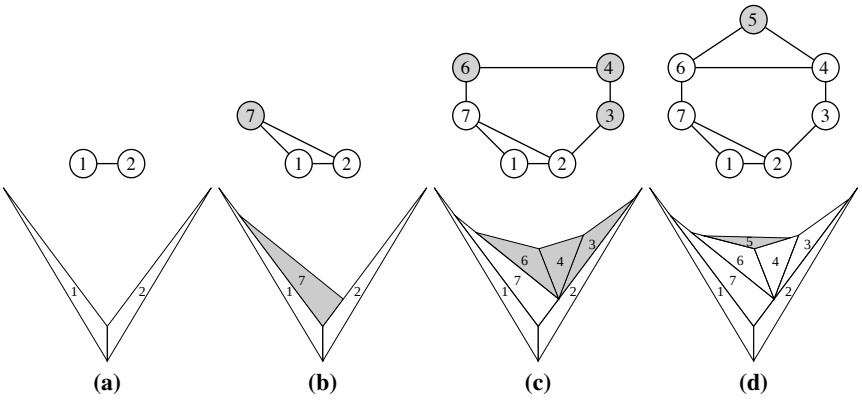

Fig. 1. Incremental construction of the TTG representation for outerplanar graphs. The shaded vertices on the top row and shaded regions on the bottom row are the ones processed at the current step.

2.1 Algorithm Overview

1. Compute an outerplanar embedding of G.
2. Compute a reverse "peeling" order of chains of vertices of G.
3. Insert region(s) corresponding to the current set of vertices in the peeling order, while maintaining a concave upper envelope.

We now look at each step in more detail. First we compute an outerplanar embedding of the graph, that is, an embedding in which all the vertices are on the outer face. For a given planar graph $G = (V, E)$, this can be easily done in linear time as follows. Let w be a new vertex and let $G' = (V', E')$, where $V' = V \cup \{w\}$ and $E' = E \cup \{(v, w)$ for all $v \in V\}$. Note that G' is planar: if it contained a subgraph homeomorphic to K_5 or $K_{3,3}$, then G would contain a subgraph homeomorphic to K_4 or $K_{3,2}$, which would imply that G was not outerplanar to begin with as these are forbidden graphs for outerplanar graphs (Theorem 11.10, [11]). We can then compute a planar embedding for G' with w on the outer face. Removing w and all its edges yields the desired outerplanar embedding, with all vertices on the outer face.

The second step of the algorithm is to compute a reverse "peeling" order of the vertices of G. Such an order is defined by peeling off one face at a time and keeping track of the set of removed vertices. If G is a single edge, the result is trivial, so we may assume that $|V| > 2$. In addition, we may assume that G is biconnected. If not, we can traverse the outer face $v_1, v_2, \ldots, v_n$. If we encounter a node $v_i = v_j, 1 < j < i$, we add a new node w_i and edges (v_{i-1}, w_i) and (w_i, v_{i+1}). The graph remains outerplanar, and we continue the traversal from v_{i+1}. This yields a biconnected graph G'. If we can construct a TTG for G', we need only remove the triangles corresponding to the added vertices to get a TTG for G.

The dual of an outerplanar graph restricted to the interior faces is a tree. We pick a face with at least one edge (v_1, v_2) on the outer face and make that the root of the tree. We then remove the faces in depth-first order. At each step, a face consists of vertices $v, u_1, u_2, \ldots, u_j, w$, where $j \geq 1$ and only the edge (w, v) is part of another face. We then remove the path $u_1, u_2, \ldots, u_j$ and continue the process until we come to the root face. We then remove the path connecting v_1 and v_2.

The third step of the algorithm is to create the touching triangles representation of G, by processing the graph using the peeling order from the second step. We begin by placing the vertices v_1 and v_2 as shown in Fig. 1(a). We then recreate each face in the reverse order in which it was removed by adding triangles corresponding to the path removed from the face. We assume that, at each step, we have the following two invariants:

1. each pair of adjacent triangles corresponding to the path from v_1 to v_2 form a concave angle
2. each triangle has part of its upper side forming part of the boundary.

This is clearly true for the first step.

Suppose the path being added consists of a single vertex w connecting to adjacent vertices v_i and v_k. Let p be the point where triangles v_i and v_k meet concavely, and let q and r be any two points of the exposed upper sides of the two triangles v_i and v_k. We can then add w as the triangle p, q, r, giving us the next face and maintaining the invariants. This is illustrated in Figures 1(b) and (d).

If the path to be added consists of multiple vertices $u_1, u_2, \ldots, u_j$, with u_1 and u_j connecting to adjacent vertices v_i and v_k, respectively, we again let the points p, q, r be as defined in the previous paragraph. We then pick points $s_1, s_2, \ldots, s_{j-1}$ so that path $q, s_1, s_2, \ldots, s_{j-1}, r$ is a concave path of line segments. We can then add this face using the triangles

$$(q, p, s_1), (s_1, p, s_2), \ldots, (s_{j-2}, p, s_{j-1}), (s_{j-1}, p, r)$$

while maintaining the invariants. Figure 1(c) shows a sample of this.

Figure 1 provides an example of the algorithm. We start with the outerplanar embedding shown in the top line of Figure 1(d), and progressively remove chains until we are left with a single edge. This is used to create the configuration shown at the bottom of Fig. 1(a). The chains are added as fans of triangles until we finish with the TTG shown at the bottom right.

The first step of this algorithm can be done in linear time as it is a slight modification of a standard planar embedding algorithm such as that by Hopcroft and Tarjan [14].

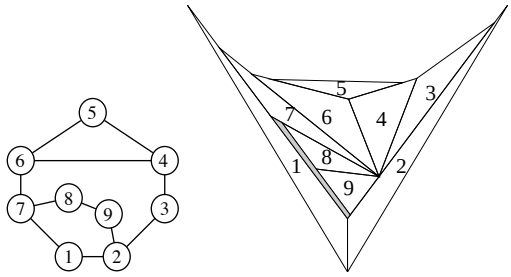

Fig. 2. Replacing a chord in an outerplanar graph with a path

The second step can also be done in linear time as computing the "peeling ordering" requires constant time per face, given the embedding of the graph from the previous step. In the third step, we record the three edges of each triangle corresponding to each processed vertex. Inserting a new chain of vertices involves finding, say, the midpoint of the exposed edges, and forming the "fan" of new triangles, all tasks which require constant time per vertex and add up to linear overall time. Thus, we have the following theorem:

Theorem 1. *A touching triangles representation can be computed in linear time for any outerplanar graph.*

Given that the above construction relies on fitting chains of triangles into smaller and smaller areas with each face, the area bounds are likely to be poor.

Corollary 1. *Any graph homeomorphic to an outerplanar graph has a touching triangles representation.*

Proof. (Sketch) Without loss of generality, we may assume that G is biconnected with an embedding such that all vertices are on the outer face except for paths of nodes connecting two nodes on the outer cycle. We then replace these interior chains by chords, yielding an outerplanar graph, and use the algorithm described above. By the construction, any chord is represented by two triangles, one of whose sides is totally within a side of the other. The shorter side can then be rotated, breaking the chord but leaving all other adjacencies intact. It is then simple to insert a fan of triangles corresponding to replacing the chord by a path of nodes. Figure 2 shows how the edge between nodes 2 and 7 in Figure 1(d) can be replaced by a chain of two nodes.

3 Grid Graphs

In this section, we show that any subgraph of a square or hexagonal grid graph can be represented by a set of touching triangles. We describe a linear time algorithm based on inserting the vertices of the graph in an outward fashion starting from an interior square/hexagon. We illustrate the algorithm with examples in Figure 3.

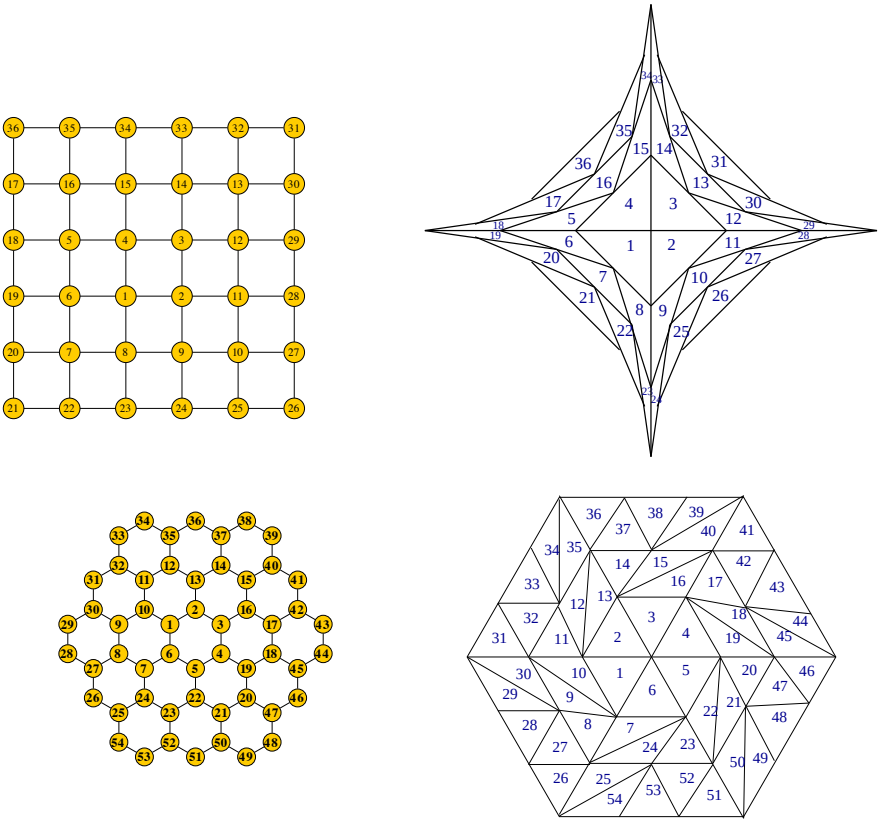

Fig. 3. Grid graphs (left) as touching triangles (right). The Hamiltonian path that visits all the vertices in the spiral order is given by the labels of the vertices.

3.1 Algorithm Overview

We first consider TTG representations for grid graphs.

1. Compute a planar embedding of G.
2. Compute a "spiral" order of the vertices of G.
3. Insert region(s), corresponding to a vertex or a path of vertices in the spiral order, while maintaining a concave upper envelope in each quadrant (in the case of square grid), or by carving out triangles out of trapezoids that correspond to the current spiral segment (in the case of hexagonal grid).

The first step of the algorithm is to compute a planar embedding of the graph, which can be done in linear time [14]. Next we compute a "spiral" order of the vertices. Such an order is defined by a Hamiltonian path which starts with the innermost face and visits all the vertices as shown in Fig. 3. Note that this is well defined for symmetric grid graphs but can be modified to handle asymmetric grid graphs and subgraphs of grid graphs.

In the case of square grids, the plane is partitioned into four quadrants and in each quadrant the spiral order introduces vertices in paths of increasing lengths $(1, 3, 5, \ldots)$.

In general these paths can be introduced recursively, provided that the upper envelope of the quadrant remains concave. The insertion of regions is similar to the process described for outerplanar graphs above.

In the case of hexagonal grids the plane is partitioned into six sectors and in each sector the spiral order introduces vertices in paths of increasing lengths $(1, 3, 5, \ldots)$. In general, these paths can be introduced directly by adding an adjacent trapezoidal region and carving it into triangles.

The above algorithms show how to construct a TTG representation for any square or hexagonal grid graph. To get a TTG representation for any subgraph, one need only remove the triangles corresponding to vertices unused in the subgraph, and adjust the remaining triangles to remove any contacts corresponding to unused edges. Thus, we have the following theorem:

Theorem 2. *A touching triangles representation can be computed in linear time for any subgraph of a square or hexagonal grid graph.*

4 Triangulations

In a triangle representation, if we require that a vertex of one triangle cannot touch the interior of the side of another, we get the special case of TTGs we call *triangulation graphs*. These representations clearly correspond to creating a triangular mesh [3, 1], allowing Steiner points within the interior of a polygon. For example, the representation in the bottom right of Fig. 3 is a triangulation graph and the representation in the top right of Fig. 3 is not.

It is easy to see that triangulation graphs form a strict subset of TTGs. For example, K_4 is a TTG but not a triangulation graph. It is also immediate that a triangulation graph has maximum degree 3, because by the definition of triangulation graphs, the vertex of one triangle cannot touch the side of another.

Lemma 1. *If G is a triangulation graph with no nodes of degree 1, G has at least 3 nodes of degree 2.*

Proof. The only triangles that can contribute to the polygon's boundary or outer face must have degree 2 in the graph, each contributing exactly 1 edge to the boundary. Since the polygon has at least 3 edges, the result follows.

Here we focus on the *filled triangulation graphs*, those whose TTG representation is filled. It is possible to fully characterize the biconnected subset of these graphs.

Theorem 3. *A biconnected graph G is a filled triangulation graph if and only if it has:*

1. *only nodes of degree 2 or 3*
2. *an embedding in the plane such that:*
   (a) *every internal node has degree 3;*
   (b) *there are at least 3 nodes of degree 2 on the boundary;*
   (c) *if there are any degree 3 nodes on the boundary, all of the degree 2 nodes cannot be consecutive; and*

(d) if the degree 2 nodes on both ends of a chain of degree 3 boundary nodes are removed, the graph remains connected.

Proof. We first prove necessity. Let G be a filled triangulation graph. Since it is biconnected, it cannot have any vertices of degree 1. Its triangulation representation yields an embedding with all internal nodes of degree 3. Lemma 1 shows we have at least 3 nodes of degree 2 on the boundary.

Suppose there are degree 3 nodes on the boundary and the degree 2 nodes are consecutive. The chain of degree 2 nodes cannot connect at a single vertex, because this would be a cut vertex. Thus, if we remove all triangles corresponding to degree 2 nodes, we would have a triangulation representation of a graph with exactly 2 vertices of degree two, which is not allowed by Lemma 1.

To finish the proof of necessity, we note that for two degree 2 triangles to disconnect the triangulation, they would have to share an interior vertex. On the other hand, if all intervening triangles on the boundary have degree 3, they can contribute nothing to the polygon boundary, so the two degree 2 must share another vertex. But then, they share a side, so there can't be any intervening degree 3 triangles.

Next, we prove sufficiency. We assume G is biconnected, all of its vertices have degree 2 or 3, and it has the specified embedding. We construct a graph G' which is a special kind of dual of G. G' contains the dual of the interior faces and edges of G. In addition, G' has a vertex for each maximal sequence of degree 3 nodes on the boundary, and a vertex for each boundary edge connecting two degree 2 nodes. These are placed in the external face of G, near the corresponding nodes or edges. These vertices are connected in a cycle of G' following the ordering induced by the boundary nodes and edges of G. Finally, for each boundary edge e of G, we add an edge from the node of G' corresponding to the interior face of G containing e to one of the vertices on the external cycle of G'. If e is adjacent to a vertex of degree 3, we connect the edge to the node of G' corresponding to the degree 3 vertex. Otherwise, we connect to the node of G' corresponding to e.

It is immediate from the construction that G' is a planar embedding of nodes and edges; all interior faces are triangles; and there is a 1-1 correspondence between faces of G' and vertices of G and between edges in G and G'. We need to show that G' is a simple graph.

As G is biconnected, G' can have no loops. Property 2(d) of the embedding implies that each interior face is connected to at most one of the nodes associated with the exterior face. The only way that multiedges could then occur would be if G' has a boundary consisting of two nodes and two edges. We know G has as least $n_2 \geq 3$ nodes of degree 2 on the boundary. If there are only degree 2 nodes on the boundary, G' has a boundary of n_2 nodes. Assume G has some degree 3 nodes on the boundary. If these nodes split into 3 or more paths, the construction creates at least 3 nodes on the boundary of G'. If not, they must split into 2 paths, since the degree 2 nodes must be separated. One group of degree 2 nodes must contain at least 2 nodes. The construction then creates one node for each group of degree 3 nodes, and at least one node for the path of more than 2 degree nodes, again given G' at least 3 boundary nodes.

As G' is simple, by using one of the algorithms (e.g, [9]) for making the edges of planar graph into line segments while retaining the embedding, we derive a triangulation representation of G, completing the proof.

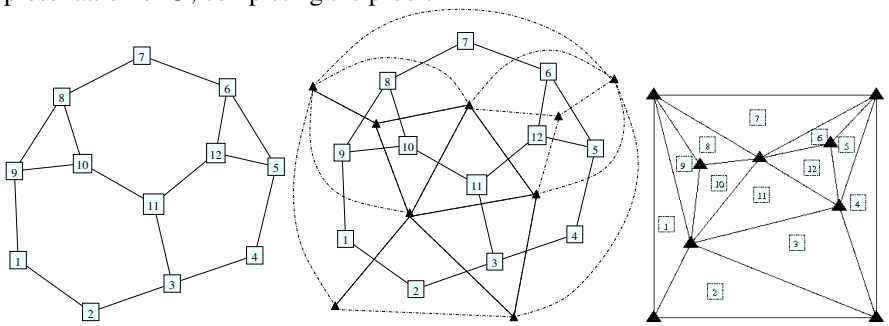

Fig. 4. Constructing a triangulation graph. **(a)** Original graph; **(b)** Creating the "dual" graph; **(c)** Straightening the edges.

Perhaps not surprisingly, the conditions of the theorem have a similar feel to those for rectangular drawings [18]. It is also not hard to see that the result can probably be derived from the duality between planar, cubic, 3-connected graphs and triangulations of the plane [19], but our proof seems more straightforward. Lastly, we note that Theorem 3 gives another proof that the hexagonal grid graphs of Section 3 are in TTG.

Figure 4 illustrates the algorithm. Figure 4(a) shows a graph satisfying the conditions of the theorem. In Figure 4(b), we have added a node for each internal face, and node on the outside for each sequence of degree 3 nodes or for each edge both of whose nodes have degree 2. This gives us a planar graph with each face having three sides and associated with a node of the original graph. Straightening the sides of the faces makes each face a triangle.

5 Necessary conditions

Thus far, we have shown that various categories of graphs are in TTG. Now, we wish to pursue some necessary conditions which will eliminate many graphs from TTG. Specifically, we show that pairs of vertices in any graph that can be represented by touching triangles must have a small common neighborhood: if a pair of vertices is connected by an edge, they cannot have more than 3 common neighbors, and if they are not directly connected, they cannot have more than 4 common neighbors. We start with some definitions.

Given triangles T_0 and T_1, pick two sides s_0 and s_1, one from each triangle, and orient the side counter-clockwise around the interior of the triangle. Extend the sides into directed lines L_0 and L_1. If the lines intersect at a unique point, the intersection is *feasible* if a non-trivial portion of s_0 lies to the right of L_1 and a non-trivial portion of s_1 lies to the right of L_0. Considering the four rays induced by the intersection, only one of the four angles corresponds to a ray pointing into the intersection followed by

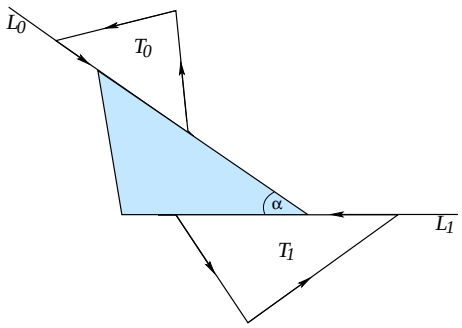

Fig. 5. A triangle T touching two other triangles T_0 and T_1. The angle α is a feasible angle of T_0 and T_1.

right turn to a ray point out. We call this a *feasible angle*. Two sides are *collinear* if the directed lines L_0 and L_1 are identical.

Lemma 2. *If a triangle T touches both T_0 and T_1, using two distinct sides, one of its angles must be a feasible angle of T0 and T1.*

Proof. If α is the angle of T determined by the two touching sides of T_0 and T_1, it immediate that α is a feasible angle. See Figure 5.

This lemma already greatly reduces the possible TTG graphs. If two triangles have no collinear sides, there can be at most nine triangles touching both of them, since any such triangle uses at least one of the feasible angles. If two sides are collinear, one triangle can touch those two sides. Any other triangles must correspond to feasible angles, and since the remaining sides of both triangles are all to the left of the two collinear sides, there can be at most 4 feasible angles. We next work at tightening these bounds.

For a node u in G, we let N_u be the nodes in G joined to u by an edge. If u and v are two nodes in a graph G, define N_{uv} as the mutual neighbors of u and v, that is, $N_{uv} = N_u \cap N_v$. Finally, define E_{uv} be the subset of edges of G induced by N_{uv}.

Theorem 4. *Let G be a TTG, and let u and v be two nodes in G joined by an edge. Then $|N_{uv}| \leq 3$ and $|E_{uv}| \leq 1$.*

Proof. Let T_u and T_v be the two triangles corresponding to nodes u and v. Since the two nodes share an edge, T_u and T_v must touch. There are basically two possibilities: one side is totally contained in the other or not.

In the first case, we have the situation represented in Figure 6. We immediately note that there can be no feasible angle associated with $\mathbf{12}$ and $\boldsymbol{ab}$. In addition, $\boldsymbol{ab}$ is to the left of both $\mathbf{23}$ and $\mathbf{31}$. On the other hand, there are feasible angles formed by $\mathbf{12}$ with $\boldsymbol{bc}$ and $\boldsymbol{ca}$. So, we only have to consider pairings of $\mathbf{23}$ and $\mathbf{31}$ with $\boldsymbol{bc}$ and $\boldsymbol{ca}$.

If point c is placed in region II, both $\boldsymbol{bc}$ and $\boldsymbol{ca}$ are to the left of $\mathbf{23}$ and $\mathbf{31}$, so there are no more feasible angles, giving a total of two.

If c is in region III, we get a new feasible angle formed by $\mathbf{31}$ and $\boldsymbol{bc}$. In this case, though, we are left with $\boldsymbol{bc}$ and $\boldsymbol{ca}$ to the left of $\mathbf{23}$, and $\mathbf{31}$ to the left of $\boldsymbol{ca}$. Thus,

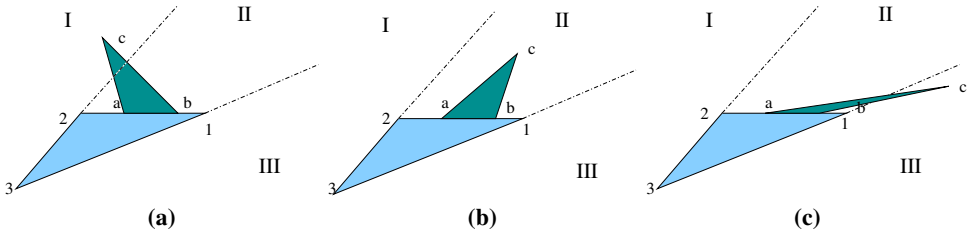

Fig. 6. Touching triangles with one side contained in the other. **(a)** Node c in region I; **(b)** Node c in region II; **(c)** Node c in region III.

we have at most three feasible points. We also note that any triangle associated with the feasible angle formed by $\mathbf{12}$ and $\boldsymbol{ca}$ cannot share an edge with any triangle of the other two feasible angles, so there can be at most one edge among the neighbors of u and v. The argument is similar if c is in region I.

If points 1 and b are identical, the same arguments hold except, in addition, we no longer have a feasible angle formed by $\mathbf{12}$ and $\boldsymbol{bc}$ because $\mathbf{12}$ is to the left of $\boldsymbol{bc}$. Thus, we have at most two mutual neighbors and no edge between them. If points 2 and a are the same, the same arguments hold. Putting these two cases together, we find that if 1 and b are identical and 2 and a are identical, there can be at most one feasible angle.

The remaining case occurs when neither shared side is contained in the other. This is the situation represented by Figure 7.

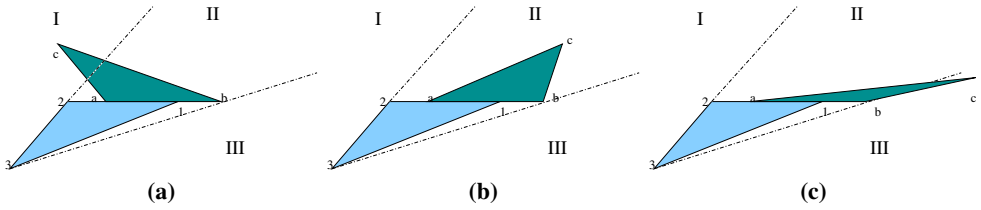

Fig. 7. Touching triangles with touching sides overlapping. **(a)** Node c in region I; **(b)** Node c in region II; **(c)** Node c in region III.

As previously, there can be no feasible angle associated with $\mathbf{12}$ and $\boldsymbol{ab}$, but now we have feasible angles formed by $\mathbf{12}$ and $\boldsymbol{ca}$, and by $\mathbf{31}$ and $\boldsymbol{ab}$. In addition, $\mathbf{12}$ is to the left of $\boldsymbol{bc}$ and $\boldsymbol{ab}$ is to the left of $\mathbf{23}$. Again, we are reduced to considering the four pairings of $\mathbf{23}$ and $\mathbf{31}$ with $\boldsymbol{bc}$ and $\boldsymbol{ca}$. If $\boldsymbol{ca}$ is to the right of $\mathbf{31}$, then $\mathbf{31}$ is to the left of $\boldsymbol{ca}$, and vice versa, so that pairing is not possible. Finally, we note that if c is in regions I or II, then $\mathbf{23}$ and $\mathbf{31}$ are to the left of $\boldsymbol{bc}$, while if c is in regions II or III, $\boldsymbol{bc}$ and $\boldsymbol{ca}$ are to the left of $\mathbf{23}$. So, if c is in region II, there are at most two feasible angles. Otherwise, there can be three but, as above, at most two of the associated triangles can touch.

We next consider what happens to the set of common neighbors if we relax the condition that there is an edge between two nodes.

Theorem 5. *Let G be a TTG, and let u and v be any two nodes in G. Then $|N_{uv}| \leq 4$ and $|E_{uv}| \leq 2$.*

Proof. The proof is similar to that of Theorem 4, and is omitted for lack of space.

6 Conclusion and Future Work

We considered the class of graphs TTG that can be represented as side-touching triangles, and showed that this includes outerplanar graphs, as well as subgraphs of square and hexagonal grids. We derived some necessary conditions for such TTG graphs, and described a complete characterization of biconnected triangulation graphs.

A complete characterization of graphs in TTG, as well as contact graphs of 4-gons and 5-gons, remains open. This is true even for the typically simpler case of filled graphs. Theorem 3 does solve the filled problem for a restricted version of TTG graphs. Can this be extended to allow for holes? Finally, the work on hexagonal contact graphs [10] gives a small ($|V| \times |V|$) area bound. Are small areas possible in the triangular or, at least, the outerplanar case?

References

1. de Berg, M., van Kreveld, M., Overmars, M.H., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer-Verlag, 2nd edn. (2000)
2. de Berg, M., Mumford, E., Speckmann, B.: On rectilinear duals for vertex-weighted plane graphs. Discrete Mathematics 309(7), 1794–1812 (2009)
3. Bern, M.: Triangulations. In: Goodman, J.E., O'Rourke, J. (eds.) Handbook of Discrete and Computational Geometry, CRC Press, 1997 (1997)
4. Bhasker, J., Sahni, S.: A linear algorithm to find a rectangular dual of a planar triangulated graph. Algorithmica 3, 247–78 (1988)
5. Brightwell, G.R., Scheinerman, E.R.: Representations of planar graphs. SIAM Journal on Discrete Mathematics 6(2), 214–229 (May 1993)
6. Buchsbaum, A.L., Gansner, E.R., Procopiuc, C.M., Venkatasubramanian, S.: Rectangular layouts and contact graphs. ACM Transactions on Algorithms 4(1) (2008)
7. de Fraysseix, H., de Mendez, P.O., Rosenstiehl, P.: On triangle contact graphs. Combinatorics, Probability and Computing 3, 233–246 (1994)
8. de Fraysseix, H., de Mendez, P.O., Rosenstiehl, P.: Representation of planar hypergraphs by contacts of triangles. In: 15th Symposium on Graph Drawing. pp. 125–136 (2007)
9. de Fraysseix, H., Pach, J., Pollack, R.: Small sets supporting Fary embeddings of planar graphs. In: 20th Symposium on Theory of Computing (STOC). pp. 426–433 (1988)
10. Gansner, E., Hu, Y., Kaufmann, M., Kobourov, S.: Optimal polygonal representation of planar graphs. In: 9th LATIN Sympoisum. pp. 417–32 (2010)
11. Harary, F.: Graph Theory. Addison-Wesley, Reading, MA (1972)
12. He, X.: On finding the rectangular duals of planar triangular graphs. SIAM Journal of Computing 22(6), 1218–1226 (1993)
13. He, X.: On floor-plan of plane graphs. SIAM Journal of Computing 28(6), 2150–2167 (1999)
14. Hopcroft, J., Tarjan, R.E.: Efficient planarity testing. Journal of the ACM 21(4), 549–568 (1974)
15. Kant, G.: Hexagonal grid drawings. In: 18th Workshop on Graph-Theoretic Concepts in Computer Science. pp. 263–276 (1992)

16. Koebe, P.: Kontaktprobleme der konformen Abbildung. Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig. Math.-Phys. Klasse 88, 141–164 (1936)
17. Liao, C.C., Lu, H.I., Yen, H.C.: Compact floor-planning via orderly spanning trees. Journal of Algorithms 48, 441–451 (2003)
18. Rahman, M., Nishizeki, T., Ghosh, S.: Rectangular drawings of planar graphs. Journal of Algorithms 50(1), 62–78 (2004)
19. Steinitz, E., Rademacher, H.: Vorlesungen über die Theorie der Polyeder. Springer-Verlag, Berlin (1934)
20. Thomassen, C.: Plane representations of graphs. In: Bondy, J.A., Murty, U.S.R. (eds.) Progress in Graph Theory, pp. 43–69. Academic Press, Canada (1984)
21. Ungar, P.: On diagrams representing maps. Journal of the London Mathematical Society 28, 336–42 (1953)