Noname manuscript No.
(will be inserted by the editor)

Convex Partitions with 2-Edge Connected Dual Graphs

Marwan Al-Jubeh · Michael Hoffmann ·
Mashhood Ishaque · Diane L. Souvaine ·
Csaba D. Tóth

Received: date / Accepted: date

Abstract It is shown that for every finite set of disjoint convex polygonal obstacles in the plane, with a total of n vertices, the free space around the obstacles can be partitioned into open convex cells whose dual graph (defined below) is 2-edge connected. Intuitively, every edge of the dual graph corresponds to a pair of adjacent cells that are both incident to the same vertex.

Aichholzer *et al.* recently conjectured that given an even number of line-segment obstacles, one can construct a convex partition by successively extending the segments along their supporting lines such that the dual graph is the union of two edge-disjoint spanning trees. Here we present counterexamples to this conjecture, with n disjoint line segments for any $n \geq 15$, such that the dual graph of any convex partition constructed by this method has a *bridge* edge, and thus the dual graph cannot be partitioned into two spanning trees.

Keywords Convex Partitions · Dual Graphs · Geometric Matchings

1 Introduction

For a finite set S of disjoint convex polygonal obstacles in the plane $\mathbb{R}^2$, a *convex partition* of the free space $\mathbb{R}^2 \setminus (\bigcup S)$ is a set C of open convex regions (called *cells*) lying in the free space such that the cells are pairwise disjoint and their closures cover the entire free space. Since every vertex of an obstacle is a reflex vertex of the free

M. Al-Jubeh, M. Ishaque, D. L. Souvaine
Department of Computer Science, Tufts University, Meford, MA, USA
email {maljub01, mishaque, dls}@cs.tufts.edu
This material is based upon work supported by the National Science Foundation under Grant No. CCF-0431027 and CCF-0830734, and Tufts Provost's Summer Scholars Program.

M. Hoffmann
Institute of Theoretical Computer Science, ETH Zürich, Switzerland
email hoffmann@inf.ethz.ch

C. D. Tóth
Department of Mathematics, University of Calgary, Calgary, AB, Canada
email cdtoth@ucalgary.ca
Supported by NSERC grant RGPIN 35586. Research done at Tufts University.

space, it must be incident to at least two cells. Let σ be an assignment of every vertex to two adjacent convex cells in C. A convex partition C and an assignment σ define a *dual graph* $D(C, \sigma)$: the cells in C correspond to the nodes of the dual graph, and each vertex v of an obstacle corresponds to an edge between the two cells assigned to v (see Fig. 1). Double edges are possible, corresponding to two endpoints of a line-segment obstacle on the common boundary of two cells.

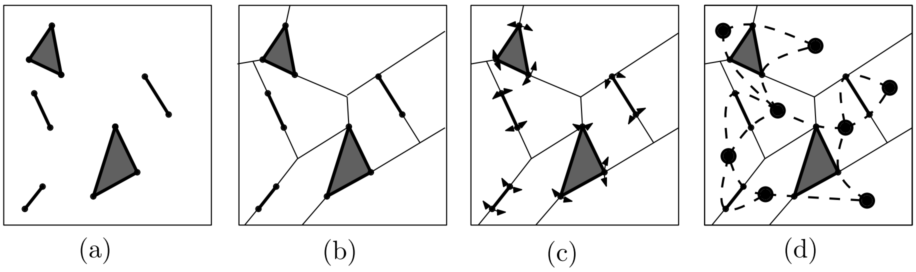

Fig. 1 (a) Five obstacles with a total of 12 vertices. (b) A convex partition. (c) An assignment σ. (d) The resulting dual graph (dashed lines).

It is straightforward to construct an arbitrary convex partition for a set of convex polygons as follows. Let V denote the set of vertices of the obstacles; and let π be a permutation on V. Process the vertices in the order π. For a vertex $v \in V$, draw a directed line segment (called *extension*) that starts from the vertex along the angle bisector (for a line-segment obstacle, the extension is collinear with the obstacle), and ends where it hits another obstacle, a previous extension, or infinity. For k convex obstacles with a total of n vertices, this naïve algorithm produces a convex partition with $n - k + 1$ cells, if no two extensions are collinear. For example, for n disjoint line segments (with $2n$ endpoints) in general position, we obtain $n + 1$ cells. If the obstacles are in general position, then each vertex v is incident to exactly two cells, lying on opposite sides of the extension emanating from v. Hence the assignment σ is unique, and the choice of permutation π completely determines the dual graph, which we denote by $D(\pi)$. We call this a Straight-Forward convex partition, and a Straight-Forward dual graph, which depends only on the permutation π of the vertices.

Results. We show instances where **no** permutation π produces a Straight-Forward dual graph $D(\pi)$ that is 2-edge connected (Section 2). This is a counterexample to a conjecture by Aichholzer *et al.* [1].

We show that for every finite set of disjoint convex polygons in the plane, with no three collinear vertices, there is a convex partition C (not necessarily Straight-Forward) and an assignment σ such that $D(C, \sigma)$ is 2-edge connected (Section 3).

We define a wide class of convex partitions, which includes all Straight-Forward convex partitions. In this class, a bridge in the dual graph is characterized by a certain simple polygon in the convex partition (a "forbidden" pattern). To build a convex partition with a 2-edge connected dual graph, we start with an arbitrary Straight-Forward convex partition and apply a sequence of "local modifications" until all forbidden patterns are eliminated. A local modification continuously deforms a simple polygon, which corresponds to a forbidden pattern. Similar continuous motion arguments have previously been used for proving combinatorial results in [2,3,16,19,21].

Motivation. A *plane matching* is a set of n disjoint line segments in the plane, which is a perfect matching on the $2n$ endpoints. Two plane matchings on the same vertex set are *compatible* if there are no two edges that cross, and are *disjoint* if there is no shared edge. Aichholzer *et al.* [1] conjectured that for every plane matching on $4n$ vertices, there is a disjoint compatible plane matching (*compatible geometric matchings conjecture*). They proved that their conjecture holds if the $2n$ segments in the matching admit a convex partition whose dual graph is the union of two edge-disjoint spanning trees, and the two endpoints of each segment corresponds to distinct spanning trees. Aichholzer *et al.* further conjectured for the $4n$ endpoints of $2n$ line segments in the plane, there is a permutation π such that the Straight-Forward dual graph $D(\pi)$ is the union of two edge-disjoint spanning trees (*two spanning trees conjecture*).

The conjecture would immediately imply that such a dual graph is 2-edge connected. Benbernou *et al.* [4] claimed that there is always a permutation π such that $D(\pi)$ is 2-edge connected—but there was a flaw in their argument [5]. Our first result shows that such permutation π does **not** always exist, and it also refutes the *two spanning trees conjecture* of Aichholzer *et al.* [1].

Related Work. Given a set of convex polygonal obstacles and a bounding box, we may think of the bounding box as a simple polygon and the obstacles as polygonal holes. Then the problem of creating a convex partition becomes that of decomposing the polygon with holes into convex parts. Convex polygonal decomposition has received considerable attention in the field of computational geometry. The focus has been to produce a decomposition with as few convex parts as possible. Lingas [18] showed that finding the *minimal convex decomposition* (decomposing the polygon into the fewest number of convex parts) is NP-hard for polygons with holes. There are approximation algorithm [15] and it is fixed parameter tractable [10]. However, for polygons without holes, minimal convex decompositions can be computed in polynomial time [9,14]– see [13] for a survey on polygonal decomposition.

While minimal convex decomposition is desirable, the number of convex parts is not the only measure of the quality of a convex partition (decomposition). In Lien's and Amato's work on approximate convex decomposition [17] with applications in skeleton extraction, the goal is to produce an approximate convex partition (where not all cells are convex) that highlights salient features. In the *equitable* convex partitioning problem, all convex cells are required to have the same value of some measure e.g. the same number of red and blues points [12], or the same area [8].

Another criterion for the quality of a convex partition might be some property of its dual graph (the definition of dual graph varies from application to application). Tan *et al.* [20] show that a Straight-Forward convex partition for sensor networks can be computed in a distributed manner, and demonstrate how it improves the performance of the geographic routing algorithms. A *leader* sensor is chosen for each cell. Communication between sensors belonging to different convex cells is routed through these leader sensors. Two leader sensors can communicate with each other if and only if they share a boundary vertex. The network of leader sensors can be modeled by the dual graph of the convex partition (although Tan *et al.* [20] did not explicitly refer to a dual graph). We believe that the communication in their model can be made fault-tolerant using a convex partition produced by our algorithm.

2 A Counterexample for the Two Spanning Trees Conjecture

Theorem 1 *For every $n \geq 15$, there are n disjoint line segments in the plane such that the* Straight-Forward *dual graph $D(\pi)$ has a bridge edge for every permutation π on the $2n$ segment endpoints.*

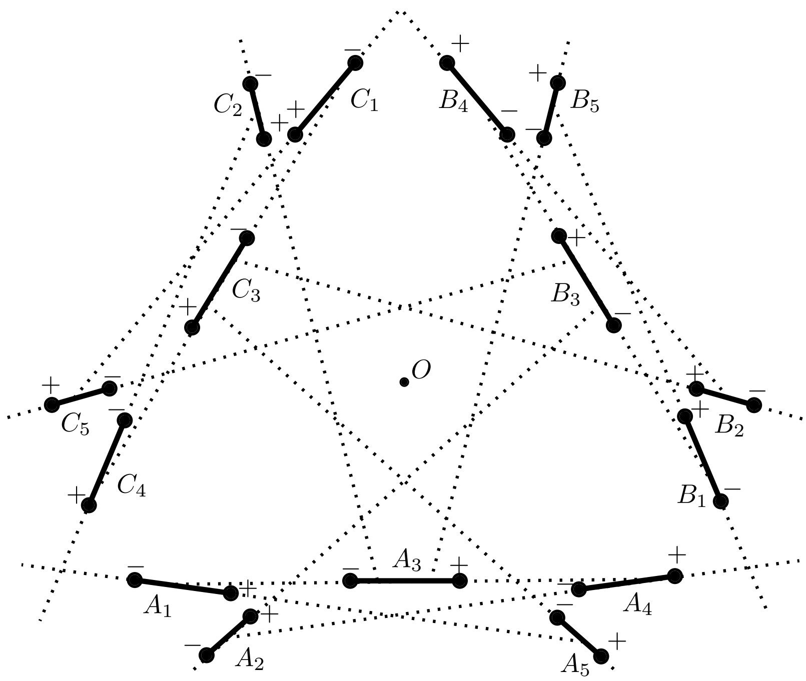

Fig. 2 A counterexample with $n = 15$ segments. Every permutation produces a Straight-Forward dual graph with a bridge edge.

Proof. We show that for the 15 line segments in Fig. 2, every permutation π produces a Straight-Forward dual graph $D(\pi)$ with a bridge edge (that is, removing this edge disconnects the dual graph). We can generate larger constructions by adding segments whose supporting lines avoid the convex hull of this configuration.

Our configuration in Fig. 2 is rotationally symmetric about the origin O. It consists of three rotationally symmetric configurations $\{A_1, A_2, \ldots, A_5\}$, $\{B_1, B_2, B_3, B_4, B_5\}$, and $\{C_1, C_2, C_3, C_4, C_5\}$, which we call *star structures*. In addition, the star structure A_i, (resp., B_i and C_i) has a reflection symmetry in the orthogonal bisector of segment A_3 (resp., B_3 and C_3). Label the endpoints of each segment X by X^- and X^+ such that the triangle OX^-X^+ has counterclockwise orientation (where O is the origin).

In Fig. 2 the dotted lines represent the arrangement of all possible extensions of the given line segments. Note that both endpoints of A_3, B_3, and C_3 lie in the interior of the convex hull of the set of all segments, and the remaining segments each have one endpoint on the convex hull. The extension from an endpoint *on the convex hull* goes to infinity or terminates at a previous extension. The extension from an endpoint in the interior of the convex hull terminates at another segment or a previous extension.

It is enough to show that any Straight-Forward convex partition has a cell incident to exactly one segment endpoint. Such a cell necessarily corresponds to a leaf node in the dual graph. We distinguish two cases.

Case 1. For at least one of A_3, B_3, and C_3, the extension from each endpoint terminates at some line segment. Assume w.l.o.g. that the extensions from endpoints A_3^- and A_3^+ terminate at segments A_1 and A_4, respectively (Fig. 2). Then the extensions from A_2^+ and A_5^- terminate at or before reaching the extensions from A_3^- and A_3^+, respectively (Fig. 2). It can be easily verified that in this case every permutation of the four endpoints $\{A_1^+, A_2^+, A_4^-, A_5^-\}$ produces a leaf in the dual graph.

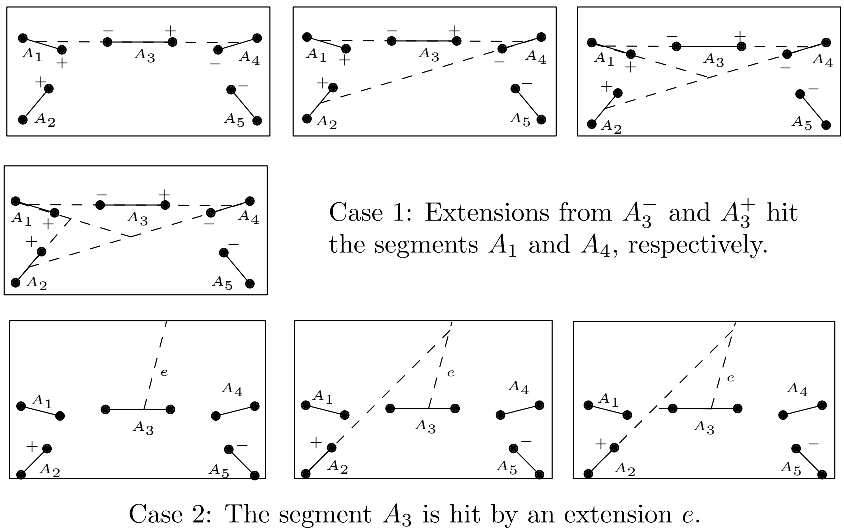

Fig. 3 Possible permutations in one star-structure of our construction.

Case 2. For A_3, B_3, and C_3, the extension from at least one endpoint does not terminate at a line segment. If the extension from endpoint A_3^- (resp. $A_3^+, B_3^-, B_3^+, C_3^-, C_3^+$) does not terminate at a line segment, then it is blocked by the extension emanating from A_2^+ (resp., $A_5^-, B_2^+, B_5^-, C_2^+, C_5^-$). In this case, we say that the extension from endpoint A_2^+ (resp., $A_5^-, B_2^+, B_5^-, C_2^+, C_5^-$) *escapes* (intuitively, it escapes from its star structure). We may assume by symmetry that the first extension in permutation π that escapes is the one emanating from C_2^+. Then the extension e from C_2^+ hits segment A_3. By our assumption, the extension from A_2^+ or A_5^- also escapes, and hits e. The extension e, an extension from A_2^+ or A_5^-, and from A_3^- or A_3^+, respectively, creates a leaf in the dual graph (Fig. 2). □

It is not essential in our construction that all obstacles are line segments. We can repeat the construction using convex polygons with arbitrarily many vertices.

Theorem 2 *For every $n \geq 15$, and integers $k_i \geq 2$, $i = 1, 2, \ldots, n$, there are n disjoint obstacles in the plane such that obstacle i is a convex polygon with k_i vertices and the* Straight-Forward *dual graph $D(\pi)$ has a bridge edge for every permutation π.*

Proof. For $n = 15$, start with the construction in Fig. 2. Label the 15 segments arbitrarily by integers 1..15. If $k_i \geq 3$, we replace segment i by a convex polygon with k_i vertices as follows. If segment I has an endpoint incident to the convex hull, then replace it by a long and skinny convex k_i-gon with $k_i - 1$ vertices in the small neighborhood of the segment endpoint on the convex hull, and one vertex at the other segment endpoint. If segment i lies in the interior of the convex hull, then replace it by a long and skinny convex k_i-gon with one vertex at each segment endpoint and $k_i - 2$ vertices in the small neighborhood of the midpoint facing the origin. The proof of Theorem 1 shows that for any permutation π of the vertices, the Straight-Forward dual graph has a leaf. If $n > 15$, we can add convex polygon such that all angle bisectors avoid the convex hull of this configuration. □

3 Constructing a Convex Partition

We showed in Section 2 that in some instances, no Straight-Forward dual graph is 2-edge connected. In this section we present an algorithm that produces a convex partition with a 2-edge connected dual graph. We will start from an arbitrary Straight-Forward convex partition, and apply a sequence of *local modifications*, if necessary, until the dual graph becomes 2-edge connected. Our local modifications will not change the number of cells. We define a class of convex partitions (Directed-Forest) that includes all Straight-Forward convex partitions and is closed under the local modifications we propose.

The basis for local modifications is a simple idea. In a Straight-Forward convex partition, extensions are created sequentially (each vertex emits a directed ray) and whenever two directed extensions meet at a *Steiner* vertex v (defined below), the earlier extension continues in its original direction, and the later one terminate (Fig. 3(a)). Here, however, we allow the two directed extensions to merge and continue as one edge in any direction that maintains the convexity of all the angles incident to v (Fig. 3(b)). Merged extensions provide considerable flexibility.

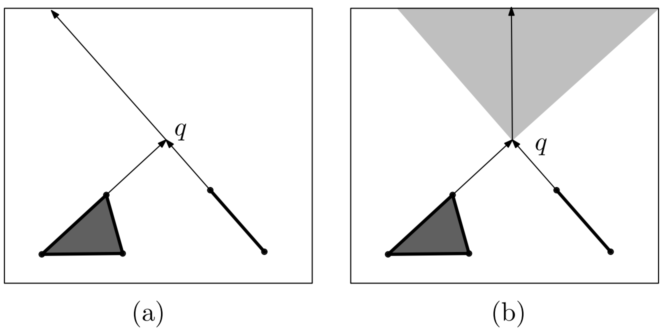

Fig. 4 (a) If two incoming extensions meet at q, the earlier extension continues in its original direction, and the later one terminates. (b) If two incoming extensions meet at q, the merged extension may continue in any direction within the opposing wedge.

Definition 1 For a given set S of disjoint convex polygonal obstacles, the class of Directed-Forest convex partitions is defined as follows (refer to Fig. 3): The free space $\mathbb{R}^2 \setminus (\bigcup S)$ is decomposed into convex cells by *directed edges* (including *directed rays* going to infinity). Denote by V the set of vertices of the polygonal obstacles in S. Each endpoint of a directed edge is either a vertex in V or a *Steiner vertex* (lying in the interior of the free space, or on the boundary of an obstacle). We require that

- every vertex in V emits exactly one outgoing edge;
- every Steiner point in the interior of the free space is incident to exactly one outgoing edge;
- no Steiner point on a convex obstacle is incident to any outgoing edge; and
- the directed edges do not form a cycle.

It is easy to see that a Straight-Forward convex partition belongs to the class of Directed-Forest convex partitions. The dual graph of a Directed-Forest convex partitions has no isolated nodes.

Proposition 1 *There is an obstacle vertex on the boundary of every cell.*

Proof. Consider a directed edge on the boundary of a cell. Follow directed edges in reverse orientation along the boundary. Since directed edges cannot form a cycle, and the out-degree of every Steiner vertex is at most one, there must be at least one obstacle vertex on the boundary of the cell. □

In a Directed-Forest, we can also follow directed edges (in forward direction) from any vertex in V to an obstacle or to infinity, since the out-degree of each vertex is always exactly one, unless the vertex lies on the boundary of an obstacle or at infinity. For connected components of extensions (directed edges), we use the concept of *extension trees* introduced by Bose *et al.* [7].

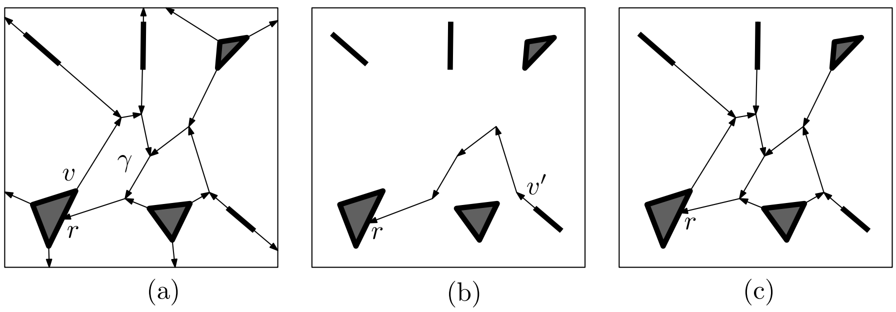

Fig. 5 (a) A convex partition formed by directed line segments. The extended path γ originates at v and terminates at r, two points on the same obstacle. The edge at v is a bridge in the dual graph, and γ is called *forbidden*. (b) A single extended-path emitted by v'. (c) A single extension tree rooted at r.

Definition 2 The **extended-path** of a vertex $v \in V$ is a directed path along directed edges starting from v and ending on an obstacle or at infinity. Its (relative) interior is disjoint from all obstacles.

Definition 3 An **extension tree** is the union of all extended-paths that end at the same point, which is called the **root** of the extension tree. The **size** of an extension tree is the number of extended-paths included in the tree.

A vertex $v \in V$ may be incident to more than two cells. It is incident to $\ell + 2$ cells if it is incident to ℓ incoming edges. In our construction, we let σ assign a vertex v of an obstacle to the two cells adjacent to the unique outgoing edge incident to v. With this convention, a bridge edge in the dual graphs $D(C, \sigma)$ can be characterized by a forbidden pattern (see Fig. 3(b)).

Definition 4 An extended-path starting at $v \in V$ is called **forbidden** if it ends at the obstacle incident to v.

A forbidden extended-path, together with the boundary of the incident obstacle, forms a simple closed curve, which encloses a bounded region.

Lemma 1 *A dual graph $D(C, \sigma)$ of a* Directed-Forest *convex partition is 2-edge connected if and only if no vertex $v \in V$ emits a forbidden extended-path.*

Proof. First we show that a forbidden extended-path implies a bridge in the dual graph. Let γ be a forbidden extended-path, starting from vertex v of an obstacle, and ending at point r on the boundary of the same obstacle (see Figures 3(b),3.2, 8). Extended-path γ together with the obstacle boundary between v and r forms a simple closed curve and partitions the free space into two regions R_1 and R_2, each of which is the union of some convex cells. Let V_1 and V_2 be the set of nodes in the dual graph corresponding to the convex cells in these regions, respectively. Point v is the only obstacle vertex along γ. If an edge e of the dual graph connects some node in V_1 to a node in V_2, then e corresponds to a vertex of an obstacle whose unique outgoing edge is part of γ. But v is the only such vertex. This implies that there is a bridge in the dual graph, whose removal disconnects V_1 from V_2.

Next we show that a bridge in the dual graph implies a forbidden extended-path. Assume that V_1 and V_2 form a partition of V in $D(C, \sigma)$ such that V_1 and V_2 are connected by a bridge edge e. The two node sets correspond to two regions, R_1 and R_2, in the free space. Let β be boundary separating the two regions. We first show that one of these regions is bounded.

Suppose for contradiction that both regions R_1 and R_2 are unbounded. Note that β must contain at least two directed edges of the convex partition that go to infinity. Since every Steiner vertex in the interior of the free space has an outgoing edge, β must contain at least two extended-paths. Hence β contains at least two vertices of some obstacles, and the adjacent outgoing edges. Thus there are at least two edges in the dual graph between the node sets V_1 and V_2, therefore, e is not a bridge edge.

Now assume without loss of generality that the region R_1 is bounded, and thus the separating boundary β is a closed curve. If we pick an arbitrary directed extension along β and follow β in reverse direction, then we arrive to a segment endpoint v. Assume that v corresponds to the bridge edge e. Then we arrive to the same segment endpoint v starting from any directed extension along β. This means that all directed edges along β are in the extended-path of v. Since β is a closed curve, the extended-path of v must end on the boundary of the obstacle incident to v, and thus it a forbidden extended-path. □

3.1 Convex Partitioning Algorithm

We construct a convex partition as follows. We first create a STRAIGHT-FORWARD convex partition, which is in the class of DIRECTED-FOREST convex partitions. Let $\mathcal{T}$ denote the set of extension trees. Each extension tree may contain one or more forbidden extended-paths. If an extension tree $t \in \mathcal{T}$ contains a forbidden extended-path γ starting from a vertex v, then we continuously deform t with a sequence of local modifications until a vertex v' of an obstacle collides with the relative interior of t (subroutine FLEXTREE(t)). At that time, t splits into two extension trees t_1 and t_2 (where t_1 contains all the extended-paths of t leading to v', and t_2 contains all the remaining extended-paths of t still leading to r). Each of these two trees (t_1 and t_2) is strictly smaller in size than t. An extension tree of size one is a straight-line extension, and cannot contain a forbidden extended-path. Since the number of extended-paths is fixed (equal to the number of vertices in V), eventually no extension tree contains any forbidden extended-path, and we obtain a convex partition whose dual graph has no bridges by Lemma 1.

For a finite set S of disjoint convex polygonal obstacles in the plane, the main loop of our partition algorithm is CREATECONVEXPARTITION(S). It calls subroutine FLEXTREE(t) for every extension tree that contains a forbidden extended-path, which in turn calls subroutine EXPAND(t, γ) for a forbidden extended path γ, as described in Section 3.2.

Algorithm 1 CREATECONVEXPARTITION(S)

Given: A set S of disjoint convex polygons having n vertices in total.
Create a STRAIGHT-FORWARD convex partition.
Let $\mathcal{T}$ be set of extension trees in the partition.
while there is an extension tree $t \in \mathcal{T}$ containing a forbidden extended-path **do**
  FLEXTREE(t)
end while

Algorithm 2 FLEXTREE(t)

Let γ be a forbidden extended-path contained in t.
while γ is still a forbidden extended-path **do**
  $(t, \gamma) =$ EXPAND(t, γ)
end while
Let $v' \in V$ be a vertex of an obstacle where the extended-path γ now terminates.
Split tree t into two extension trees t_1 and t_2. Subtree t_1 consists of the extended-paths that now terminate at v'. Subtree t_2 consists of the extended-paths that terminate at the original endpoint of γ.

3.2 Local Modifications: EXPAND(t, γ)

Consider a forbidden extended-path γ contained in an extension tree $t \in \mathcal{T}$. Path γ starts from a vertex $v \in V$, and ends at a root r lying on the boundary of the obstacle

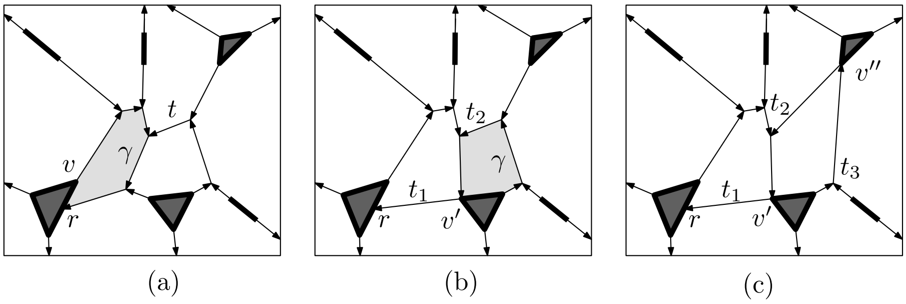

Fig. 6 (a) An extension tree t with a forbidden extended-path. (b) After deforming and splitting t into two trees, t_2 contains a forbidden extended-path. (c) Deforming and splitting t_2 eliminates all forbidden extended-paths.

$s \in S$ incident to v. Let P be the simple polygon formed by path γ and a portion of the boundary of s between v and r such that s is disjoint from the interior of P.

We continuously deform the boundary of P, together with the extension tree t, until it collides with a new vertex $v' \in V$ that is not incident to s. We deform P in a sequence of local deformations, or *step*s.

What is a local deformation step? Each step involves either one edge or two consecutive edges of the polygon P. The *vertices of* P are v, r and the Steiner points where P has an interior angle different from 180°. Steiner vertices where P has an interior angle of 180° are considered interior points of edges of P. During the deformation of an edge of P, one endpoint of the edge moves along a straight line trajectory and the other endpoint is fixed. Each step of the deformation will

(i) increase the interior of the polygon P,
(ii) keep r a vertex of P, and
(iii) maintain a valid Directed-Forest convex partition.

The third condition implies, in particular, that every cell has to remain convex. Since the interior of P is increasing, some cells in the exterior of P (and adjacent to P) will shrink—we maintain that every cell adjacent to P has a nonempty interior.

Note that an edge of P may contain Steiner vertices of the convex partition. At each such Steiner point, an extended path starting from either the interior of or the exterior of P joins the forbidden extended-path along the boundary of P (Fig. 3.2(a)). In the area swept by the deforming edges, the adjacent edges of the convex subdivision are either truncated (in the exterior of P) or extended (in the interior of P), while maintaining a valid Directed-Forest convex partition at all times (Fig. 3.2(b-c)). In particular, the Steiner points along the deforming edges also move continuously

The continuous deformation of the boundary of P can be discretized based on the sweep-line paradigm of Bentley and Ottmann [6]. Since the deforming edges follow algebraic trajectories (one endpoint is fixed, and the other moves along a straight line), we can maintain a queue of *combinatorial changes*, which are: (1) two Steiner points along a deforming edge merge; (2) A Steiner point along a deforming edge splits into several Steiner points; (3) A deforming edge of P becomes collinear with an adjacent directed edge of the convex partition.

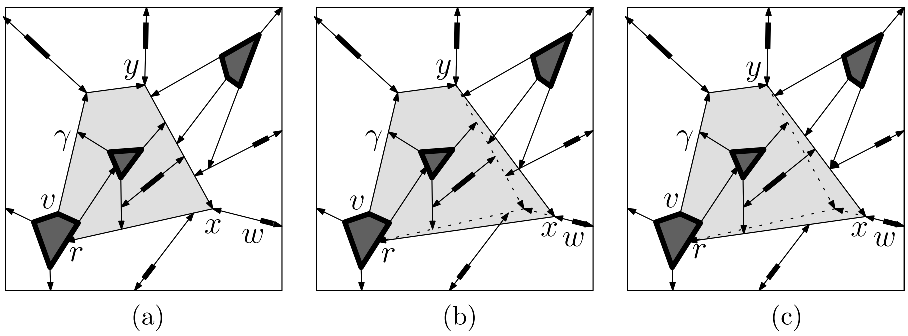

Fig. 7 (a) An extension tree t with a forbidden extended-path. (b) A local deformation step stretches edges xy and xr of P such that x continuously modes along the edge xw towards w. (c) We update the extended paths in the area swept by the deforming edges.

Where to perform a local deformation step? The polygon P is modified either at a convex vertex x on the convex hull of P or at a certain reflex vertex x'. The vertex x or x' is calculated at the start of each local deformation step.

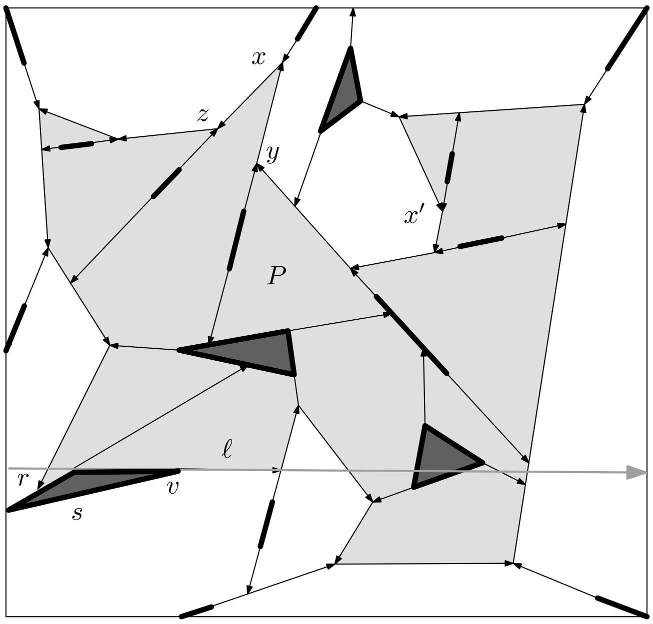

Fig. 8 Polygon P corresponding to a forbidden extended-path $v, \ldots, r$; convex vertex x; inflexible edges xy and xz; reflex vertex x'.

Consider the edge of the obstacle s that is incident to the point v, and is part of the boundary of the polygon P. Let ℓ be the supporting line through this edge. The obstacle s lies completely in one of the closed halfplanes bounded by ℓ (since s is convex). Let x be a vertex of P furthest away from the supporting line ℓ in the other halfplane. Clearly, x is a convex vertex of P (interior angle less than 180°), otherwise it will not be the furthest. The goal is to expand the polygon P by modifying the edges xy and xz incident to x. Imagine grabbing the vertex x and pulling it away from the polygon P stretching the edges xy and xz. But this expansion can only occur if both the edge xy and xz are *flexible*. An edge of P is *inflexible* if there is a convex cell in the interior of P such that part of the edge lies on the boundary of the cell and the cell has an angle of 180° at one endpoints of the edge. Since x is a convex vertex, the edge xy or xz can be inflexible if and only if some convex cell has an angle of 180° at y or z, respectively (Fig. 8).

In the case that at least one of the edges incident to x is inflexible, local modification of P takes place at a reflex vertex x'. Assume w.l.o.g. that xy is inflexible. Then y must be a reflex vertex of P (every inflexible edge of P is incident to a reflex vertex). Starting with the reflex vertex y, move along the boundary of P in the direction away from x. Let x' be the first reflex vertex encountered such that one of the edges incident to x' is flexible. By Proposition 2 (below), there is always one such vertex x'.

Proposition 2 *If vertex x is incident to an inflexible edge, then there is a reflex polygonal chain along P that includes this inflexible edge and terminates at a reflex vertex x' of P that has exactly one flexible edge.*

Proof. Let xy be an inflexible edge incident to x. Consider the maximal reflex polygonal chain λ along P starting from x and containing xy. Each internal vertex of λ is reflex, and it is incident on at most one cell in the interior of P which has an angle of 180° at that vertex. Since each cell is convex, its boundary can overlap with at most one edge of λ. That is, a cell at each internal vertex of λ is responsible for making at most one edge of λ inflexible. Since a polygonal path has one fewer internal vertices than edges, λ has at least one flexible edge. Let x' be the first internal vertex of λ incident to a flexible edge along λ. □

How to perform a local deformation step? Local deformation of P takes place either at a convex vertex x (Cases 1 and 2), or at a reflex vertex x' (Case 3). Since the number of cells in the convex partition must remain the same, it is necessary to check for the collapse of a cell in the exterior of P (Case 4).

Case 1. Both edges xy and xz of P incident to x are flexible, and there is an edge wx in the opposing wedge of $\angle yxz$. Fig. 3.2(a). Then continuously move x along xw towards w while stretching the edges xy and xz.

Case 2. Both edges xy and xz of P incident to x, are flexible, and there is no edge in the opposing wedge of $\angle yxz$. Fig. 3.2(b). Let ℓ_x be the line parallel to ℓ passing through x, and let w be a neighbor of x on the opposite side of ℓ_x. Assume that z and w are on the same side of the angle bisector of $\angle yxz$. Then split x into two vertices x_1 and x_2. Now x_1 remains fixed at x and x_2 moves continuously along xw towards w stretching the edge x_2z.

Case 3. At least one edge incident to x is inflexible; then there is a reflex vertex x' such that edge $x'z'$ is inflexible, and $x'y'$ is flexible. Fig. 3.2(c). Continuously move x' along $x'z'$ towards z' while stretching the edge $x'y'$.

Case 4. The next combinatorial change during stretching some edge ab to position ab', where vertex b continuously moves along segment bb', would collapse a cell c in the exterior of P. Fig. 3.2(d). Then the interior of triangle $\Delta abb'$ is disjoint from obstacles,

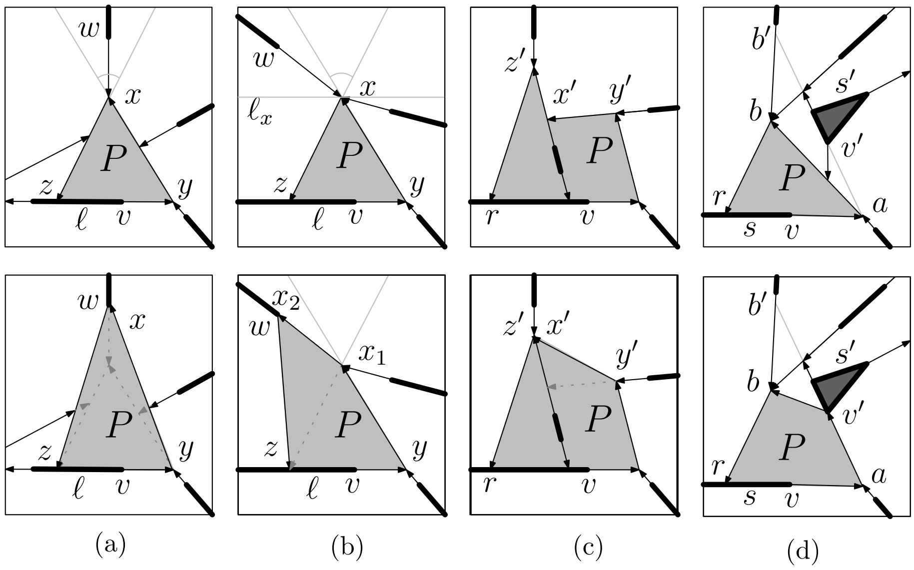

Fig. 9 Three local operations: (a) Convex vertex x, incoming edge w in the wedge. (b) Convex vertex x, no incoming edge in the wedge. (c) Reflex vertex x. (d) The case of a collapsing cell.

and a side of some obstacle lies in segment ab' and is adjacent to cell c (cf. Proposition 3 below). Let $v' \in ab'$ be the vertex of this obstacle that lies closer to a. Stretch edge ab of P into the path (a, v', b).

When to stop a local deformation step? Continuously deform one or two edges of P, at either a convex vertex x or a reflex vertex x', until one of the following conditions occurs:

– an angle of a convex cell interior to P or an angle of P becomes 180°;
– two vertices of the polygon P collide;
– one of the edges of P collides either in its interior or at its endpoint with a vertex v' of an obstacle;
– the next combinatorial change in the deformation would collapse a cell in the exterior of P.

Since a local deformation step does not always terminate in a collision with an obstacle vertex v', the subroutine FlexTree(t) decides at the end of each step whether more local modifications are needed.

3.3 Correctness of the Algorithm

We prove that we can eliminate all forbidden extended-paths and obtain a Directed-Forest convex partition with a 2-edge connected dual graph. Let t be a extension tree, containing a forbidden extended-path γ starting from $v \in V$ and ending at root r. First we show that in Expand(t, γ), the four cases cover all possibilities.

Proposition 3 *If the next combinatorial change while deforming some edge ab to position ab', where b continuously moves along segment bb', would collapse a cell c in the exterior of P, then the interior of triangle $\Delta abb'$ is disjoint from obstacles, and and a side of some obstacle lies in segment ab' and is adjacent to cell c.*

Proof. A continuous deformation of ab to ab', where b' moves along segment bb', sweeps triangle $\Delta abb'$. Hence the interior of $\Delta abb'$ is disjoint from obstacles, and if any extended-path intersects the interior of $\Delta abb'$, then it is part of the extension tree t. Assume that cell $c \in C$ would collapse if ab reaches position ab'. By Proposition 1, there is a vertex $v' \in V$ on the boundary of cell c, and so v' must lie on the line segment ab'. Since every extended-path in interior of triangle $\Delta abb'$ is part of t, no extension in $\Delta abb'$ terminates at v'. Hence the two edges along the boundary of c incident to v' are the extension emitted by v' and a side of the obstacle s' containing v'. It follows that segment ab' contains a side of obstacle s'. □

Proposition 4 *Every step* Expand(t, γ) *preserves a* Directed-Forest *convex partition and it also preserves the number of cells.*

Proof. It is clear that the continuous deformations in Cases 1-3 maintain a valid Directed-Forest with the same number of cells. Consider Case 4. Denoting by $v' \in ab'$ the vertex of this side that lies closer to a, we stretch edge ab of P into the path (a, v', b). By the choice of v', edge av' does not contains any side of obstacles. Edge $v'b$ traverses triangle $\Delta abb'$, so it does not any side of obstacles, either. By Proposition 3, a side of some obstacle s' lies in segment ab' and is adjacent to cell c, and so it cannot collapse in this step of Expand(t, γ). A valid Directed-Forest convex partition is preserved with the same number of cells. □

Lemma 2 *The subroutine* FlexTree(t) *modifies an extension tree $t \in \mathcal{T}$, with a forbidden extended-path γ, in a finite number of* Expand(t, γ) *steps until an obstacle vertex $v' \in V$ appears along γ.*

Proof. FlexTree(t) repeatedly calls Expand(t, γ) for a forbidden extended-path γ. We associate an integer $\text{count}(t, \gamma)$ to t and γ and show that Expand(t, γ) either deforms t to collide with an obstacle $s \neq s'$ or $\text{count}(t, \gamma)$ strictly decreases. This implies that FlexTree(t) terminates in a finite number of steps.

Let k denote the size of t (i.e., the number of extended-paths in t). Then t has at most $k - 1$ Steiner vertices in the free space, since each corresponds to the merging of two or more extended-paths. Let k_{ex} be the number of Steiner vertices of t in the exterior of P, let r_P be the number of vertices of P, let f_P be the number of flexible edges of P, and let m_p be the number of extended-paths in t that enter vertex x of P from the exterior of P. Then let $\text{count}(t, \gamma) = 2k \cdot k_{\text{ex}} + r_P + f_P + 2m_P$. Recall that a Steiner vertex where P has an internal angle of $180°$ is not a vertex of P. The vertices of P are v, r and Steiner vertices in the interior of the free space where P has a non-straight internal angle, hence $r_p, f_P, m_P < k$.

Consider a sequence of Expand(t, γ) steps where t does not collide with an obstacle. Since in Case 4, a vertex $v' \in V$ appears in the relative interior of t, we may assume that only Case 1–3 are applied. Case 1–3 expand the interior of polygon P, and the directed edges in the exterior of P are not deformed. Hence k_{ex} never increases, and it decreases if P expands and reaches a Steiner point in the exterior of P.

Now consider a sequence of EXPAND(t, γ) steps where k_{ex} remains fixed and Case 4 does not apply. Then m_P can only decrease in Case 1–3. Case 2 initially introduces a new edge of P (increasing r_P and f_P by one each) but it also decreases m_P by at least one. Case 1 and 3 never increase r_P or f_P. In Case 1–3, the deformation step terminates when an interior angle of a convex cell within P becomes 180° (and an edge becomes inflexible, decreasing f_P) or an interior angle of P becomes 180° (and P loses a vertex, decreasing r_P). In both events, $r_P + f_P$ decreases by at least one. Therefore, $\text{count}(t, \gamma) = 2k \cdot k_{\text{ex}} + r_P + f_P + 2m_P$ strictly decreases in every step EXPAND(t, γ), until the relative interior of t collides with an obstacle. □

Theorem 3 *For every finite set of disjoint convex polygonal obstacles in the plane, there is a convex partition and an assignment σ such that the dual graph $D(C, \sigma)$ is 2-edge connected. For k convex polygonal obstacles with a total of n vertices, the convex partition consists of $n - k + 1$ convex cells.*

Proof. The convex partitioning algorithm first creates a STRAIGHT-FORWARD convex partition for the given set of disjoint polygonal obstacles. For k disjoint obstacles with a total of n vertices, it consists of $n - k + 1$ convex cells. The extensions in the convex partition can be represented as a set of extension trees $\mathcal{T}$. We showed in Lemma 1 that there is a bridge in the dual graph iff some extension tree contains a forbidden extended-path. Subroutine FLEXTREE(t) splits every extension tree t containing a forbidden extended-path into two smaller trees. (The extended-paths in t are distributed between the two resulting trees.) An extension tree that consists of a single extended-path is a straight-line extension, and cannot be forbidden (a straight-line extension emitted from a vertex of an obstacle cannot hit the same obstacle, since each obstacle is convex.) Therefore, after at most $|V|/2$ calls to FLEXTREE(t), no extended-path is forbidden, and so the dual graph of the convex partition is 2-edge connected. □

4 Conclusion

We have shown that for every finite set of convex polygonal obstacles, there is a convex partition with a 2-edge connected dual graph. We have presented a polynomial time algorithm for constructing the dual graph. It is straightforward to implemented our algorithm in $O(n^4 \log n)$ time for obstacles with a total of n vertices. Each EXPAND(t, γ) operation involves stretching two edges of a simple polygon, and can be implemented in $O(n \log n)$ time. By Lemma 2, each FLEXTREE(t) requires $O(n^2)$ calls to EXPAND(t, γ). Finally, each FLEXTREE(t) increases splits the extension tree t into two, where the number of extension trees is at most n—the number of vertices.

For comparison, the STRAIGHT-FORWARD convex partition for any permutation π can be computed in $O(n \log^2 n)$ time [11]. For a permutation π where the vertices with rightward pointing angle bisectors come before the vertices with leftward pointing angle bisectors, a STRAIGHT-FORWARD convex partition can be computed in $O(n \log n)$ time by two line sweeps.

Tan *et al.* [20] computes STRAIGHT-FORWARD convex partition in a distributed manner, which makes the algorithm suitable for sensor networks. However, it remains to be seen whether the algorithm to produce convex partitions with 2-edge connected dual graphs presented in this paper could be modified to work in a distributed manner. A related question is how to support efficient insertion and deletion of convex polygonal obstacles. In other words: how *local* the local modifications really are?

References

1. O. Aichholzer, S. Bereg, A. Dumitrescu, A. García, C. Huemer, F. Hurtado, M. Kano, A. Márquez, D. Rappaport, S. Smorodinsky, D. Souvaine, J. Urrutia, D. Wood, Compatible geometric matchings, *Comput. Geom.* **42** (2009), pp. 617–626.
2. A. Andrzejak, B. Aronov, S. Har-Peled, R. Seidel, and E. Welzl, Results on k-sets and j-facets via continuous motion, in: *Proc. 14th Sympos. Comput. Geom.*, 1998, ACM Press, pp. 192–199.
3. T. F. Banchoff, Global geometry of polygons I: The theorem of Fabricius-Bjerre, *Proc. AMS* **45** (1974), pp. 237–241.
4. N. Benbernou, E. D. Demaine, M. L. Demaine, M. Hoffmann, M. Ishaque, D. L. Souvaine, and C. D. Tóth, Disjoint segments have convex partitions with 2-edge connected dual graphs, in: *Proc. Canadian Conf. Comput. Geom.*, 2007, pp. 13–16.
5. N. Benbernou, E. D. Demaine, M. L. Demaine, M. Hoffmann, M. Ishaque, D. L. Souvaine, and C. D. Tóth, Erratum for "Disjoint segments have convex partitions with 2-edge connected dual graphs", in: *Proc. Canadian Conf. Comput. Geom.*, 2008, p. 223.
6. J. L. Bentley and T. A. Ottmann, Algorithms for reporting and counting geometric intersections, *IEEE Transactions on Computers* **C-28** (9) (1979), 643-647.
7. P. Bose, M. E. Houle, and G. T. Toussaint, Every set of disjoint line segments admits a binary tree, *Discrete Comput. Geom.* **26** (3) (2001), pp. 387–410.
8. J. G. Carlsson, B. Armbruster, and Y. Ye, Finding equitable convex partitions of points in a polygon efficiently, *ACM Transactions on Algorithms* (2009), to appear.
9. B. Chazelle and D. P. Dobkin, Optimal Convex Decompositions, *Comput. Geom.* **2** (1985), pp. 63–133.
10. M. Grantson and C. Levcopoulos, A fixed parameter algorithm for the minimum number convex partition problem, in: *Proc. Japan Conf. Discrete Comput. Geom. (Tokyo, 2004)*, vol. 3746 of LNCS, Springer, 2005, 83–94.
11. M. Ishaque, B. Speckmann, and C. D. Tóth. Shooting permanent rays among disjoint polygons in the plane, in: *Proc. 25th Sympos. Comput. Geom.*, 2009, ACM Press, pp. 51–60.
12. A. Kaneko and M. Kano, Perfect partitions of convex sets in the plane, *Discrete Comput. Geom.* **28** (2) (2002), pp. 211–222.
13. M. Keil, Polygon decomposition, in: *Handbook of Computational Geometry (J.-R. Sack and J. Urrutia, Eds.)*, Elsevier, Amsterdam, 2000, pp. 491–518.
14. M. Keil and J. Snoeyink, On the time bound for convex decomposition of simple polygons, *Internat. J. Comput. Geom. Appl.* **12** (2002), pp. 181–192.
15. C. Knauer and A. Spillner, Approximation algorithms for the minimum convex partition problem, in *Proc. SWAT*, vol. 4059 of LNCS, Springer, 2006, pp. 232–241.
16. D. W. Krumme, E. Rafalin, D. L. Souvaine, and C. D. Tóth, Tight bounds for connecting sites across barriers, *Discrete Comput. Geom.* **40** (3) (2008), pp. 377–394.
17. J.-M. Lien and N. M. Amato, Approximate convex decomposition of polygons, *Comput. Geom.* **35** (1-2) (2006), pp. 100–123.
18. A. Lingas, The power of non-rectilinear holes, in: *Proc. 9th ICALP*, vol. 140 of LNCS, 1982, Springer, pp. 369–383.
19. I. Streinu, Pseudo-triangulations, rigidity and motion planning, *Discrete Comput. Geom.* **34** (4) (2005), 587–635.
20. G. Tan and M. Bertier and A-M. Kermarrec, Convex partition of sensor networks and its use in virtual coordinate geographic routing, in: *Proc. INFOCOM*, 2009, IEEE, pp. 1746–1754.
21. C. D. Tóth, Guarding disjoint triangles and claws in the plane. *Comput. Geom.* **25** (1-2) (2003), 51–65.