Convex Partitions with 2-Edge Connected Dual Graphs

Marwan Al-Jubeh*
Dept. of Computer Science
Tufts University, Medford
maljub01@cs.tufts.edu

Michael Hoffmann
Inst. Theor. Comp. Sci.
ETH Zürich, Switzerland
hoffmann@inf.ethz.ch

Mashhood Ishaque*
Dept. of Computer Science
Tufts University, Medford
mishaq01@cs.tufts.edu

Diane L. Souvaine*
Dept. of Computer Science
Tufts University, Medford
dls@cs.tufts.edu

Csaba D. Tóth
Department of Mathematics
University of Calgary
cdtoth@ucalgary.ca

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 a counterexample to this conjecture, which consists of 16 disjoint line segments, 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. Counterexamples of arbitrarily larger sizes can be constructed similarly.

Questions about the dual graph of a convex partition are motivated by the still unresolved conjecture about disjoint compatible geometric matchings by Aichholzer *et al.*. It has application in the design of fault-tolerant wireless networks in the presence of obstacles (e.g. tall buildings in a city).

[*] Partially supported by NSF grants CCF-0431027 and CCF-0830734, and Tufts Provost's Summer Scholars Program.

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 *cell*s) 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 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.

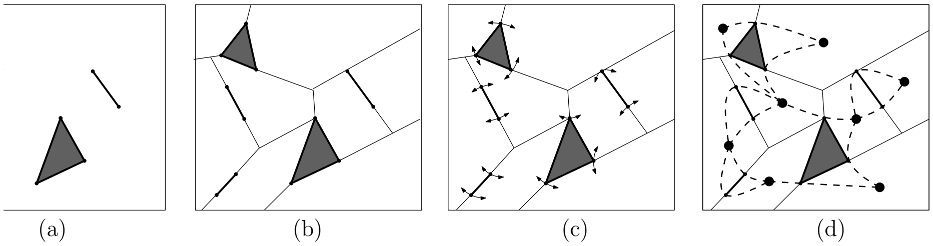


Figure 1: (a) Five obstacles. (b) A convex partition. (c) An assignment σ. (d) The resulting dual graph.

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 (the bounding box). For k convex obstacles, where each obstacle i has $d_i \geq 2$ vertices, this naïve algorithm produces a convex partition with $1 + \sum_{i=1}^{k}(d_i - 1)$ cells, if no two extensions are collinear. For example, for n disjoint line segments 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 and so we have $D(C_\pi, \sigma) = D(\pi)$. We call this a STRAIGHT-FORWARD convex partition, and a STRAIGHT-FORWARD dual graph, which depends only on the permutation π of the vertices V.

1.1 Our 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 there is a convex partition (not necessarily STRAIGHT-FORWARD) and an assignment that produces a 2-edge connected dual graph (Section 3).

1.2 Motivation

Disjoint compatible matching conjecture. 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 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] (mentioned above).

Fault-tolerant wireless networks. In today's information age, users of mobile phones, wireless data services, and GPS devices etc, have come to expect ubiquitous access to information–anywhere, anytime. The providers of these services try to meet the needs of their demanding customers or risk losing business. But it is difficult to provide uninterrupted service in the presence of obstacles such as tall buildings in a city (e.g. Manhattan) that may impede the signals from satellites or cell towers. Underground subways pose a similar problem. One possible solution is to install inexpensive radio devices on the boundaries of these obstacles, which can communicate with each other and provide connectivity to the users. We can model the obstacles as convex polygons in the plane, and place radios on the boundaries of these obstacles. Our goal is to partition the service areas between radios such that (i) each radio serves clients in two designated convex areas, and (ii) between any two clients there are *at least two* disjoint sequences of radios such that consecutive radios are not occluded by obstacles. The purpose of the second condition is to provide fault-tolerance so that clients can still have access even when one of the radios responsible for the area fails. The communication network of these radio devices is exactly the dual graph defined in this paper. Our results show that if the radios are placed at the corners of all obstacles, then we can designate two convex regions to each radio with properties (i) and (ii). The result can also be applied to sensor networks in the sense of Tan *et al.* [9].

1.3 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 simple 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 [16] showed that finding the *minimal convex decomposition* (decomposing the polygon into the fewest number of convex parts) is NP-hard for polygons with holes. However, for polygons without holes, minimal convex decompositions can be computed in polynomial time [8, 13]–see [12] for a survey on polygonal decomposition.

While minimal convex decomposition is desirable, it is not the only criterion for the *goodness* of a convex partition (decomposition). In fact, the measure of the quality of a convex partition can be specific to the application domain. In Lien's and Amato's work on approximate convex decomposition [15] with applications in skeleton extraction, the goal is to produce an approximate (not all cells are convex) convex partition 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 [11], or the same area [7] (with application to vehicle routing).

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.* [9] 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.* [9] 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 Counterexample for Two Spanning Trees Conjecture

Theorem 1 *For every $n \geq 16$, there are n disjoint line segments in the plane such that the dual graph $D(\pi)$ has a bridge edge for every permutation π.*

Proof. We show that for the 16 line segments in Fig. 2, every permutation π produces a STRAIGHT-FORWARD dual graph $D(\pi)$ with a bridge edge (removing this edge disconnects the dual graph). Our construction consists of three rotationally symmetric copies of a configuration with 5 segments $\{A_1, A_2, \ldots, A_5\}$, which we call a *star structure*. The purpose of segment D_0 is just to make the number of segments even. We can generate larger constructions using a similar strategy. In Fig. 2 the dotted lines represents the arrangement of all possible extensions of the given line segments. Denote the right endpoint of a segment by '$+$' and the left endpoint by '$-$'. The set of all possible permutations can be described in terms of only two cases by focusing on the star structure A.

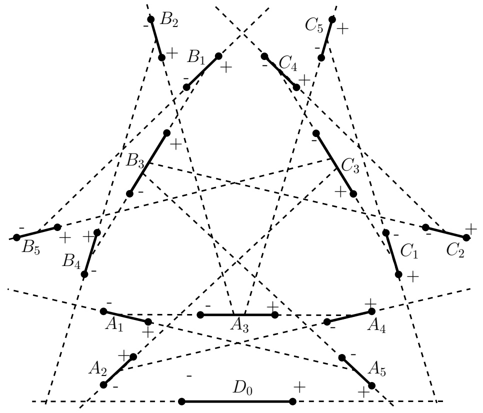


Figure 2: Counterexample with 16 segments

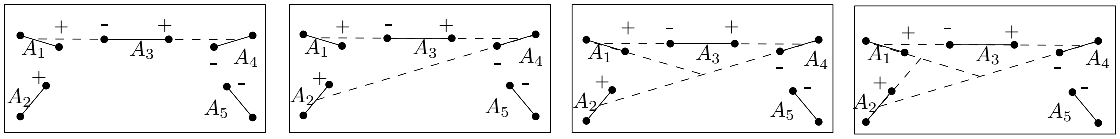


Figure 3: Case 1: Extensions from A_3^- and A_3^+ hit the segments A_1 and A_4, respectively.

Case 1. The extensions from endpoints A_3^- and A_3^+ hit the segments A_1 and A_4, respectively; i.e. the extensions from endpoints A_2^+ and A_5^- terminate either at the extensions from endpoints A_3^- and A_3^+, respectively or earlier (Fig. 3). 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 bridge in the dual graph. The same reasoning applies to the structures B and C because of symmetry. Therefore, to avoid a bridge edge in the dual graph, there must be at least one endpoint in each star structure whose extension goes beyond the structure. Since when two extensions meet in a STRAIGHT-FORWARD convex partition, one of the extensions must continue in a straight line, at least one of these three endpoints will have its extension hit a segment (A_3, B_3 or C_3) in a different structure. Assume w.l.o.g that the segment hit by such an extension is A_3.

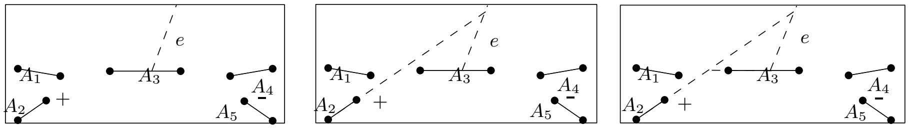


Figure 4: Case 2: The segment A_3 is hit by an extension e

*Case 2.*The segment A_3 is hit by an extension e from either the endpoint B_2^+ or C_5^-. Then an extension from either A_3^- or A_3^+ will hit e creating a bridge in the dual graph. □

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 terminates. 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. 5(a)). Merged extensions provide considerable flexibility.

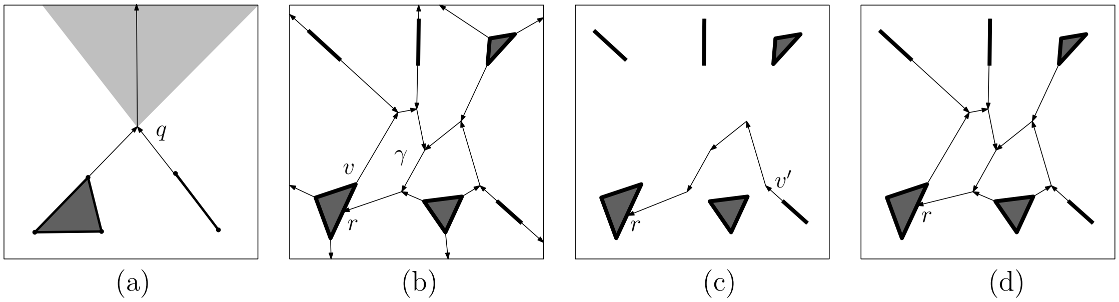


Figure 5: (a) If two incoming extensions meet at q, the merged extension may continue in any direction within the opposing wedge. (b) A convex partition formed by directed line segments. The extended path γ originates at v and terminates at r, two points on the same obstacle. The edge at v is a bridge in the dual graph, and γ is called *forbidden*. (c) A single extended-path emitted by v'. (d) A single extension tree rooted at r.

Definition 1 *For a given set S of disjoint obstacles, the class of* DIRECTED-FOREST *convex partitions is defined as follows (refer to Fig. 5): The free space $\mathbb{R}^2 \setminus (\bigcup S)$ is partitioned into convex cells by* directed edges *(including* directed rays *going to infinity). Each endpoint of a directed edge is either a vertex of S 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 (a vertex of an obstacle) 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 clear that a STRAIGHT-FORWARD convex partition belongs to the DIRECTED-FOREST class.

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

Proof. Consider a directed edge on the boundary of a convex cell, and 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.* [6].

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. 5(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 every* 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 5(b),6, 7). 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.

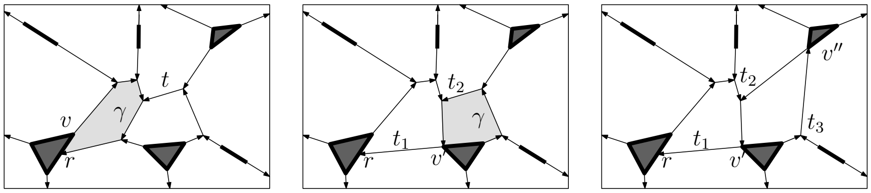
Suppose for contradiction that both regions R_1 and R_2 are unbounded. Note that β must contain at least two directed edges of the convex partition that go to infinity. Since every Steiner vertex in the interior of the free space has an outgoing edge, β must contain at least two extended-paths. Hence 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. □

Corollary 1 *An extension tree with its root at infinity cannot contain 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, then we continuously deform t with a sequence of local modifications until a vertex 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.



(a) (b) (c)

Figure 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.

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

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 $s \in S$ incident to v. Path γ together with the portion of the boundary of s between v and r bounds a simple polygon P that does not contain s in its interior. We continuously deform the boundary of P, together with extension tree t, until it collides with a new vertex $v' \in V$ that is not incident to s. Similar continuous motion arguments have been used for proving combinatorial properties in [2, 3, 14]. We deform P in a sequence of local modifications, or *step*s. Each step involves two adjacent 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. 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. Also, since the interior of P is increasing, some cells in the exterior of P (and adjacent to P) will shrink—we ensure that all cells adjacent to P have a nonempty interior.

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 reflex vertex x' (with special properties). These vertices x and x' are calculated at the start of each local deformation step.

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). Then the convex vertex x is an extremal 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 that has an angle of 180° at one of the two 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. 7).

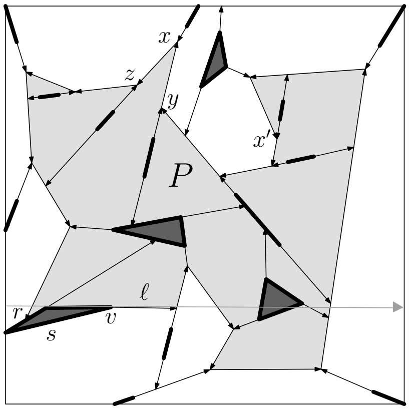


Figure 7: Polygon P; supporting line ℓ; vertex x; inflexible edges xy and xz; reflex vertex x'; forbidden extended-path as boundary of P from v to r.

In the case when 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 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 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. It is not difficult to verify that there is always one such vertex x' (Proposition 1).

Proposition 1 *If x is incident to an inflexible edge, then there is a polygonal chain along P of length ≥ 1 that includes this inflexible edge and terminates at a reflex vertex x' of P that has exactly one flexible edge.*

How to perform a local deformation step? Local deformation of P takes place either at a convex vertex x (case 1 and 2), or at a reflex vertex x' (case 3). Since the number of cells in 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. 8(a). Then continuously move x along xw towards w while stretching the edges xy and xz.

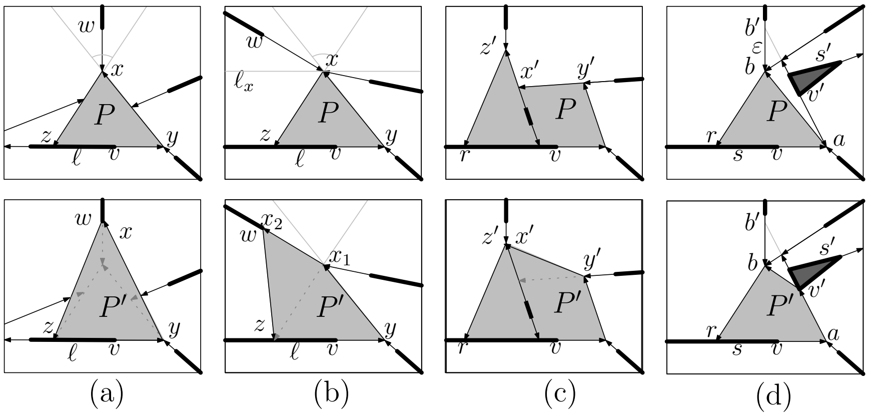


Figure 8: Three local operations: (a) Convex vertex x, incoming edge w in the wedge. (b) Convex vertex x, no incoming edge in the wedge. (c) Reflex vertex x. (d) The case of a collapsing cell.

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. 8(b). Let ℓ_x be a line parallel to ℓ passing through x, and let w be the 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. 8(c). Continuously move x' along $x'z'$ towards z' while stretching the edge $x'y'$.
Case 4. A further $\varepsilon > 0$ stretching of some edge ab to position ab', where vertex b continuously moves along segment bb', would collapse a cell in the exterior of P. Fig. 8(d). Then the triangle $\Delta abb'$ lies in the free space and ab' contains a side of an obstacle $s' \neq s$. 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).

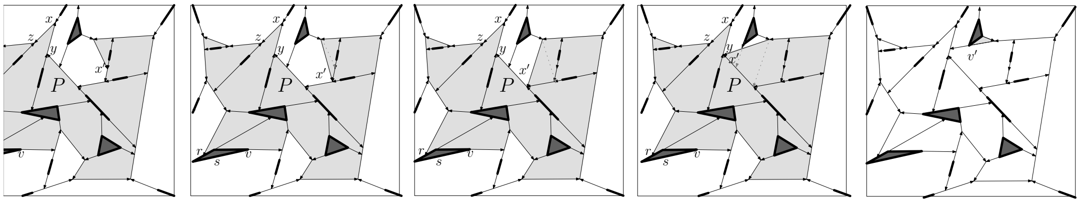


Figure 9: Polygon P is continuously deformed until an obstacle vertex appears on the boundary.

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 condition occurs:

- an angle of a convex cell interior to P or an interior 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;
- a further $\varepsilon > 0$ 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 $\text{EXPAND}(t, \gamma)$, the four cases cover all possibilities.

Proposition 2 *If a further $\varepsilon > 0$ deformation of some edge ab to position ab', where b continuously moves along segment bb', would collapse a cell in the exterior of P, then the triangle $\Delta abb'$ lies in the free space and segment ab' contains a side of an obstacle $s' \neq s$.*

Proof. A continuous deformation of ab to ab', where b' moves along segment bb', sweeps triangle $\Delta abb'$. Hence the interior of this triangle cannot contain any obstacles. Assume that cell $c \in C$ would collapse if ab reaches position ab'. By Observation 1, there is a vertex $v' \in V$ on the boundary of cell c, and so v' must lie on the segment ab'. Note that no extended-path can reach v' from the triangle $\Delta abb'$. Hence the only two edges along the boundary of c incident to v' are the extension emitted by a side of the obstacle s' containing v'. It follows that segment ab' contains a side of obstacle s'.

It remains to be shown that $s' \neq s$ (that is, v and v' are vertices of distinct obstacles). In cases 1-2, $b = x$ is the convex vertex of P that lies furthest from the supporting line ℓ, and b moves continuously away from ℓ. Therefore both b and b' are in the open halfplane bounded by ℓ, and so edge ab' cannot contain an edge of the obstacle s. In case 3, $b = x'$ is a reflex vertex of P and it moves continuously along a reflex chain along the boundary of P between x' and x (c.f., Proposition 1). Since x is the furthest point from the supporting line ℓ of s, the reflex chain between x and x' is separated from s by a line. Segment ab' lies in the convex hull of the reflex chain, and so it cannot contain a side of s. □

Proposition 3 $\text{EXPAND}(t, \gamma)$ *(i) increases the interior of polygon P, (ii) keeps r as a vertex of P, and (ii) maintain a valid* DIRECTED-FOREST *convex partition. Furthermore,* $\text{EXPAND}(t, \gamma)$ *modifies only directed edges of the extension tree t.*

Proof. It is easy to check cases 1-4 that $\text{EXPAND}(t, \gamma)$ satisfies the requirements (i)–(iii), and it does not modify any edge of extension trees other than $t \in \mathcal{T}$. □

Next, we show that $\text{FLEXTREE}(t)$ terminates in a finite number of steps.

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

Proof. $\text{FLEXTREE}(t)$ repeatedly calls $\text{EXPAND}(t, \gamma)$ for a forbidden extended-path γ. We associate an integer $\text{count}(t, \gamma)$ to t and γ and show that $\text{EXPAND}(t, \gamma)$ either deforms t to collide with an obstacle $s \neq s'$ or it strictly decreases $\text{count}(t, \gamma)$. This implies that $\text{FLEXTREE}(t)$ 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 directed edges in t that are incident to 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 steps $\text{EXPAND}(t, \gamma)$ 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 cases 1-3 are applied. Cases

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 steps $\text{EXPAND}(t,\gamma)$ where k_{ex} remains fixed and case 4 does not apply. Then m_P can only decrease in cases 1-3. Case 2 initially introduces a new edge of P (increasing r_P and f_P by one) but it also decreases m_P by at least one. Cases 1 and 3 never increase r_P or f_P. In cases 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 $\text{EXPAND}(t,\gamma)$, until the relative interior of t collides with an obstacle. □

Theorem 2 *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 $\text{FLEXTREE}(t)$ 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 $\text{FLEXTREE}(t)$, no extended-path is forbidden, and so the dual graph of the convex partition is 2-edge connected. □

Corollary 2 *For every finite set of disjoint line segments 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 n disjoint line segments, the convex partition consists of $n+1$ convex cells.*

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. Such a dual graph can be used in the design of fault-tolerant wireless or sensor networks in the presence of polygonal obstacles. We have presented a polynomial time algorithm for constructing the dual graph. Our algorithm can easily be implemented in $O(n^3)$ time for obstacles with a total of n vertices, however we believe that the runtime can be substantially reduced by a more careful analysis and by using specialized data structures. For comparison, for any given permutation π the STRAIGHT-FORWARD convex partition C_π can be computed in $O(n \log^2 n)$ time [10], and if the permutation π is not specified, then a STRAIGHT-FORWARD convex partition can be computed in $O(n \log n)$ time by a sweep-line algorithm (in a left-to-right sweep followed by a right-to-left sweep).

The question regarding the existence of convex partitions whose dual graph can be edge-partitioned into two disjoint spanning trees remains open. An affirmative answer to this question will settle the disjoint compatible matching conjecture by Aichholzer *et al.*. An interesting related problem is whether there always exists a convex partition that has a bi-connected (2-vertex connected) dual graph.

Tan *et al.* [9] 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, and D. R. Wood, Compatible geometric matchings, *Comput. Geom.* **42** (6-7) (2009), 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 SCG*, 1998, ACM, pp. 192–199.

[3] T. F. Banchoff, Global geometry of polygons I: The theorem of Fabricius-Bjerre, *Proc. AMS* **45** (1974), 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. CCCG*, 2007, 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. CCCG*, 2008, 223.

[6] 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), 387–410.

[7] 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.

[8] B. Chazelle and D. P. Dobkin, Optimal Convex Decompositions, *Comput. Geom.* **2** (1985), 63–133.

[9] M. B. Guang Tan and A.-M. Kermarrec, Convex partition of sensor networks and its use in geographic routing, in: *Proc. INFOCOM*, 2009, to appear.

[10] M. Ishaque, B. Speckmann, and C. D. Tóth. Shooting permanent rays among disjoint polygons in the plane, in: *Proc. 25th SCG*, 2009, ACM, to appear.

[11] A. Kaneko and M. Kano, Perfect partitions of convex sets in the plane, *Discrete Comput. Geom.* **28** (2) (2002), 211–222.

[12] M. Keil, Polygon decomposition, in: *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, Eds. Elsevier, Amsterdam, 2000, 491–518.

[13] M. Keil and J. Snoeyink, On the time bound for convex decomposition of simple polygons, *Internat. J. Comput. Geom. Appl.* **12** (2002), 181–192.

[14] 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), 377–394.

[15] J.-M. Lien and N. M. Amato, Approximate convex decomposition of polygons, *Comput. Geom.* **35** (1-2) (2006), 100–123.

[16] A. Lingas, The power of non-rectilinear holes, in: *Proc. 9th ICALP*, 1982, Springer, 369–383.