Minimum Weight Convex Quadrangulation of a Constrained Point Set

Thomas Fevens Henk Meijer David Rappaport

Department of Computing and Information Science

Queen's University, Kingston, Ontario K7L 3N6 Canada

Email: {fevens,henk,daver}@qucis.queensu.ca

December 12, 1997

Abstract

A convex quadrangulation with respect to a point set S is a planar subdivision whose vertices are the points of S, where the boundary of the unbounded outer face is the boundary of the convex hull of S, and every bounded interior face is convex and has four points from S on its boundary. A minimum weight convex quadrangulation with respect to S is a convex quadrangulation of S such that the sum of the Euclidean lengths of the edges of the subdivision is minimised. In this paper, we will present a polynomial time algorithm to determine whether a set of points S admits a convex quadrangulation if S is constrained to lie on a fixed number of nested convex polygons, where the time complexity is polynomial in the cardinality of S. This algorithm can also be used to find a minimum weight convex quadrangulation of the point set.

1 Introduction

There are many problems for which it is necessary to find a numerical solution of a complicated system of differential equations. To find such a numerical solution, the finite element method [6] [19] is often employed. With this method, the domain over which the solution is sought is divided into small pieces (finite elements) determined by data points sampled from the domain. This subdivision, often generated automatically or semi-automatically, is termed a *mesh*. Typically, the mesh is comprised of triangular finite elements, but for some situations, such as for interpolating scattered data [10] [11] as well as for some finite element method applications [3] [7], it is preferable to use finite elements that are quadrangles (quadrilaterals) instead of triangles. Such a quadrangulation with respect to a point set S is a planar subdivision whose vertices are the points of S, where the boundary of the unbounded outer face is the boundary of the convex hull of S, and every bounded interior face has four points from S on its boundary.

Some of the earliest results with respect to quadrangulations were for orthogonal polygons, i.e., polygons whose boundaries are parallel to two orthogonal axes. Note that we will consider a polygon to include its interior. This work was actually motivated by its application to locating $n/4$ guards to cover (i.e., watch over) the interior of an empty polygon, i.e., a polygon with no points in its interior, with n vertices. Sack and Toussaint [17] showed that they could partition a star-shaped orthogonal polygon into convex quadrangles in $O(n)$ time. Kahn, Klawe and Kleitman [8] gave an existential proof that arbitrary simple orthogonal empty polygons can be partitioned into convex quadrangles. They also showed that any orthogonal polygon with n vertices and h holes can be decomposed into $n/2+h-1$ convex quadrangles. Following up on this, Sack [16], Sack and Toussaint [18], and independently Lubiw [12] each gave constructive proofs of Kahn, Klawe and Kleitman's existential proof. The time complexity of their algorithms was $O(n \log n)$. Also, Lubiw showed, using a fairly general computational model, that any algorithm used to decompose an orthogonal polygon with holes into convex quadrangles requires $O(n \log n)$ time.

Work has also been done on finding minimum weight (minimum "ink") quadrangulations where the sum of the Euclidean lengths of the line segments of the quadrangulation is minimised. With regards to the minimum weight quadrangulation of empty polygons, Keil and Sack [9] and Lubiw [12] showed that a minimum weight convex quadrangulation, where all the quadrangles are convex, of a simple orthogonal polygon could be constructed in $O(n^4)$ time and $O(n^2)$ space using a dynamic programming algorithm. By modifying Keil and Sack's algorithm, Conn and O'Rourke [4] [5] constructed minimum weight convex quadrangulations (MWCQs) of simple polygons and orthogonal polygons in $O(n^3 \log n)$ time and $O(n^3)$ space. If the simple polygon does not admit a convex quadrangulation, a value of infinity is returned for the minimum weight value. On a related problem, Conn and O'Rourke [4] [5] also constructed minimum weight convex quadrangulations of empty convex polygons in $O(n^3)$ time. Note that this $O(n^3)$ time is the same time complexity as for the best known algorithm for a minimum weight triangulation for a convex polygon.

Conversely, relatively little published work has been done with regards to convex quadrangulations of planar point sets. There have been some heuristic algorithms for generating "well-shaped" quadrangulations, e.g., the work of Bose and Toussaint [2] using algorithms based on a serpentine triangulation of the point set.

At the 4th International Workshop on Algorithms and Data Structures, WADS '95, Toussaint posed the problem of deciding whether one can "decide if a set of points admits a quadrangulation where every quadrangle is convex?", noting that this question was first raised by Joe Mitchell at the MSI Workshop in Stonybrook in 1993 [20].

In this paper, we extend the work of finding MWCQs to planar point sets, and thus present a partial answer to the question posed by Toussaint. To explore this area of research, we turn to the work of Anagnostou and Corneil on the Minimum Weight Triangulation problem with respect to point sets arranged on the boundaries of nested convex hulls [1]. Using their dynamic programming techniques, and some ideas from Conn and O'Rourke [4] [5], we develop a MWCQ algorithm for point sets.

2 MWCQ of a Constrained Point Set

The approach of Anagnostou and Corneil for the Minimum Weight Triangulation problem is centered on a dynamic programming algorithm based on finding optimal partitions of "legal polygons" of varying sizes (cardinality) defined from the point set, S [1]. At each stage of the algorithm, the optimal triangulation for a legal polygon is constructed from the best of all possible partitions of the polygon into legal subpolygons of smaller cardinality. Only one optimal solution is then kept for each legal polygon.

Anagnostou and Corneil extended their method to the Minimum Weight Triangulation problem on a point set constrained to non-crossing straight lines. A more efficient solution to this latter problem was also presented by Meijer and Rappaport [13].

2.1 Preliminaries

Before we can discuss the partition of legal polygons and other aspects of the dynamic programming algorithm, we describe some terminology used in this paper. Most of the definitions are borrowed directly from Anagnostou and Corneil [1]. Also, most of the geometric terminology is standard and details can be found in [14]. Note that in the following development, any vertex within or on the boundary of a legal polygon, or polygon, comes from the input point set S.

Let E^2 denote the Euclidean space in two dimensions. We assume that any given point set S in E^2 is in general position, i.e., no three points are collinear. Given a set of points, S, the convex hull of S, denoted as $CH(S)$, is defined to be the minimum area convex polygon enclosing S.

Now, some definitions with regard to the nested convex hulls.

Definition 2.1 *Let $CH_i(S)$, the nested convex hulls of S, be defined by*

$$\begin{aligned} CH_1(S) &= CH(S), \\ CH_i(S) &= CH\left(S - \bigcup_{j=1}^{i-1} (\textit{vertices on the boundary of } CH_j(S))\right) \qquad \forall i \geq 2. \end{aligned}$$

For a point $x \in S$ on the boundary of the convex hull $CH_i(S)$, we let $h(x)$ be the index of the convex hull such that $h(x) = i$. Also, we let $\hbar$ denote the total number of nested convex hulls, which is the maximum i such that $CH_i(S)$ has at least one vertex on its boundary.

Definition 2.2 *For a point $x \in S$, define $\alpha(x)$ to be the set of all points $s \in S$ such that the open line segment (s, x) does not intersect $CH_{h(x)}(S)$.*

Definition 2.3 *If $x, y \in S$, a line segment (x, y) is called legal with respect (l.w.r.) to y if $x \in \alpha(y)$.*

For example, see figure 1. In this example, $\hbar = 4$, and the different convex hulls are labelled. For the point p, $h(p) = 3$ and $\alpha(p) = \{b, c, d, e, f, g, j, k, l\}$.

Our dynamic programming algorithm operates on a polynomial number of subproblems in the form of legal polygons. The rest of this section develops the notion of legal polygons and culminates in the definition of a legal polygon and its associated quadrangulation.

Definition 2.4 *A path $(p_1, p_2, ..., p_m)$ is a simple planar straight-line graph (see [15]) with vertex*

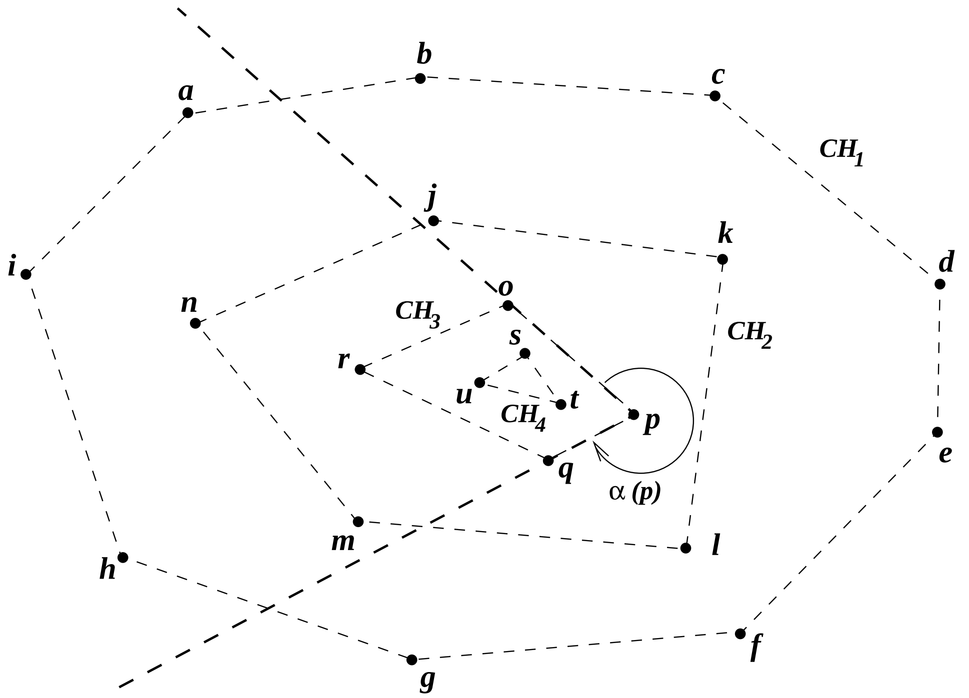

Figure 1: Nested convex hulls and related definitions. Adapted from [1].

set $\{p_1, p_2, ..., p_m\}$ *and line segment set* $\{(p_i, p_i + 1) : i = 1, ..., m - 1\}$*. The length of a path is the number of its line segments.*

Definition 2.5 *A legal path is a path with a starting vertex on* $CH(S)$*, and each vertex,* x*, on the path is connected to the next vertex,* y*, by a line segment* (x, y) *that is l.w.r. to* y*.*

Note that the maximum length of a legal path is $\hbar$. We call the last vertex of a legal path the *end point* of the path. Further, we define $p + q$ as meaning the concatenation of the paths p and q, when q starts with the end point of p. For example, if $p = (p_1, p_2, ..., p_k)$ and $q = (p_k, p_{k+1}, ..., p_l)$, then $p + q = (p_1, p_2, ..., p_k, p_{k+1}, ..., p_l)$.

Definition 2.6 *Two distinct legal paths* $p = (p_1, ..., p_k)$ *and* $q = (q_1, ..., q_l)$ *are defined to be compatible if the following four conditions hold:*

(1) p *and* q *have different end points,* $p_k \neq q_l$*;*

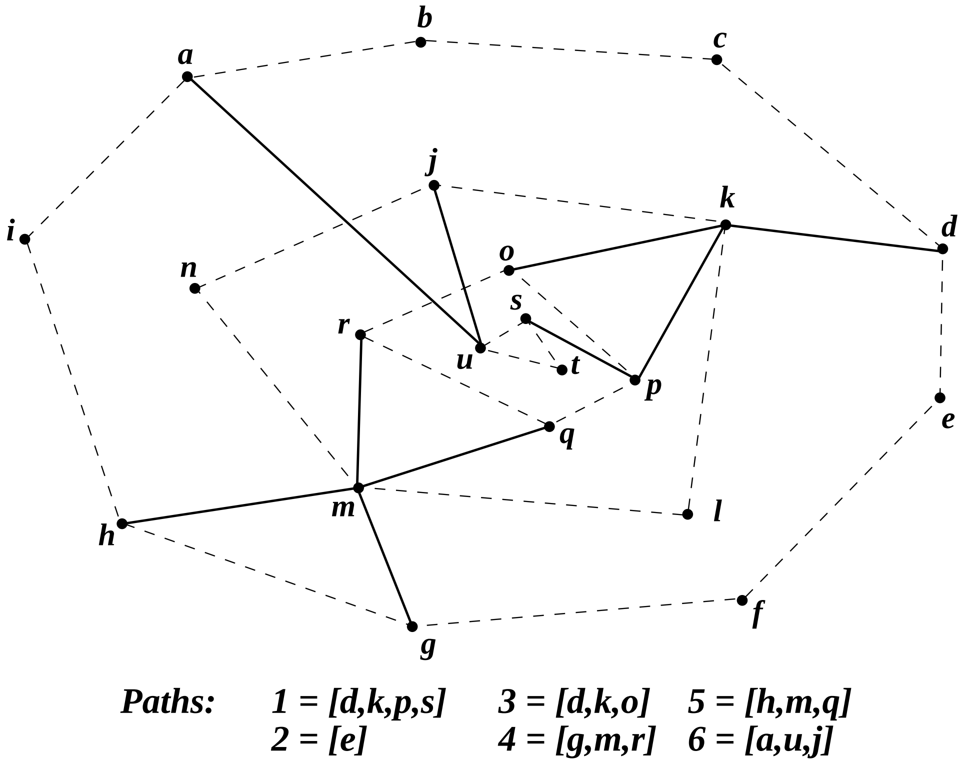

Figure 2: Examples of Paths. Adapted from [1].

(2) The open line segment connecting the two end points of p and q either does not intersect p or q or is wholly contained in either p or q;

(3) There exists a partition of the plane into C, a simple curve, and C^+, C^-, two half spaces, such that p is contained in $C \cup C^+$, and q is contained in $C \cup C^-$. That is, p and q may overlap, but they do not cross; and

(4) If p and q share a vertex t, then the path from p_1 to t is identical to the path from q_1 to t.

See figure 2 for an example to help clarify the above definitions. In this example, paths 1, 2, 3, 4 and 5 are legal. Path 6 is not legal since edge (u, j) is not l.w.r. to j. Paths 1 and 2 are not compatible because edge (e, s) intersects path 1. Paths 4 and 5 are not compatible since they cross at m.

We now consider quadrangulations of a polygon.

Definition 2.7 *Consider a simple polygon P. Let S be a point set in the plane including the vertices of the boundary of P, δP. A quadrangulation of P is a planar subdivision whose vertices are $S \cap P$ and where every face interior to the polygon boundary is bounded and is a quadrangle.*

A quadrangulation of an empty polygon would be a special case where $S \cap P$ only includes the vertices of P.

For the remainder of the paper, a quadrangulation of a polygon is always assumed to be a quadrangulation with respect to a point set S.

The dynamic programming algorithms will be based on finding the quadrangulation of legal polygons, which we now define.

Definition 2.8 *Consider two compatible paths $p = (p_1, p_2, ..., p_k)$ where $(k \geq 1)$, and $q = (q_1, q_2,$ $..., q_l)$ where $(l \geq 1)$. These paths may be combined to form a legal polygon $P(p, q)$ as follows. We define $\delta P(p, q)$ to be the boundary of $P(p, q)$. Therefore, a legal polygon $P(p, q)$ is the closed region of the plane bounded by*

$$\begin{aligned}
\delta P(p,q) \quad = \quad & (p_k, p_{k-1}, ..., p_1) + \\
& (\textit{the possibly empty clockwise part of the boundary of } CH(S) \textit{ from } p_1 \textit{ to } q_1) + \\
& (q_1, q_2, ..., q_l) + (q_l, p_k).
\end{aligned}$$

Definition 2.9 *Consider a legal polygon $P(p, q)$ and a legal path t with an end point at $z \in S \cap P(p, q)$. We consider t to be completely within $P(p, q)$ if $t \subseteq P(p, q)$.*

Note that if p and q share a vertex a, and t_1 and p_1 are the vertices of t and p on $CH(S)$, then the path from t_1 to a is identical to the path from p_1 to a. Also, a legal polygon is not strictly a

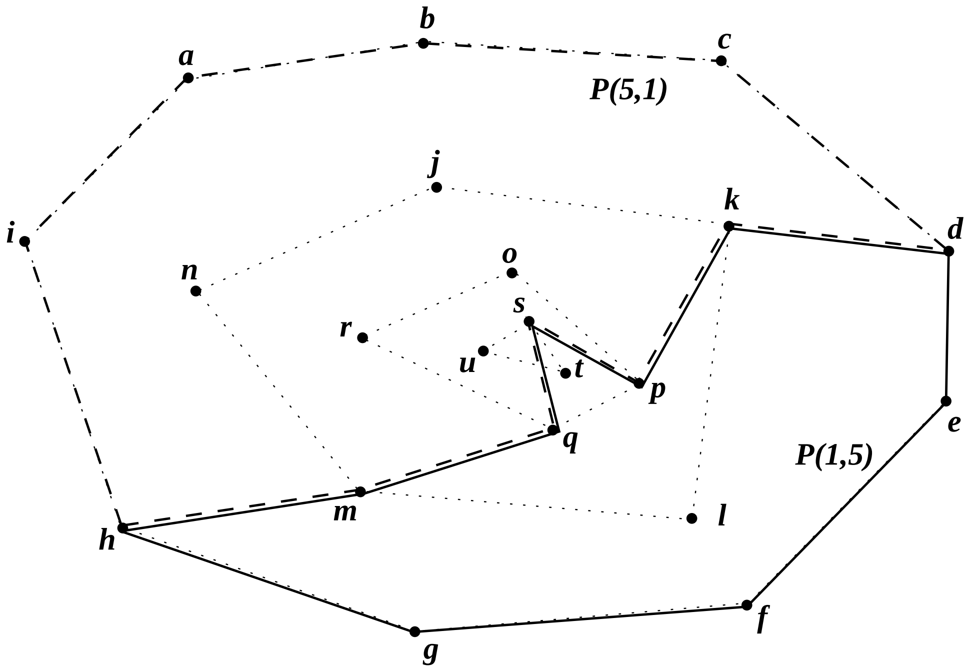

Figure 3: Example legal polygons. The path numbering used is from figure 2. Adapted from [1].

simple polygon since some edges of p and q may be identical, as shown in figure 4. Therefore, the definition of a quadrangulation of a polygon does not strictly apply to legal polygons.

Definition 2.10 *Consider a legal polygon $P(p,q)$. Let $\delta P'$ denote the boundary of the interior of $P(p,q)$. Note that $\delta P'$ is the boundary of a polygon P'. Either P' is a simple polygon, or we can treat it as a simple polygon by considering any common edges of p and q as being distinct (see figure*

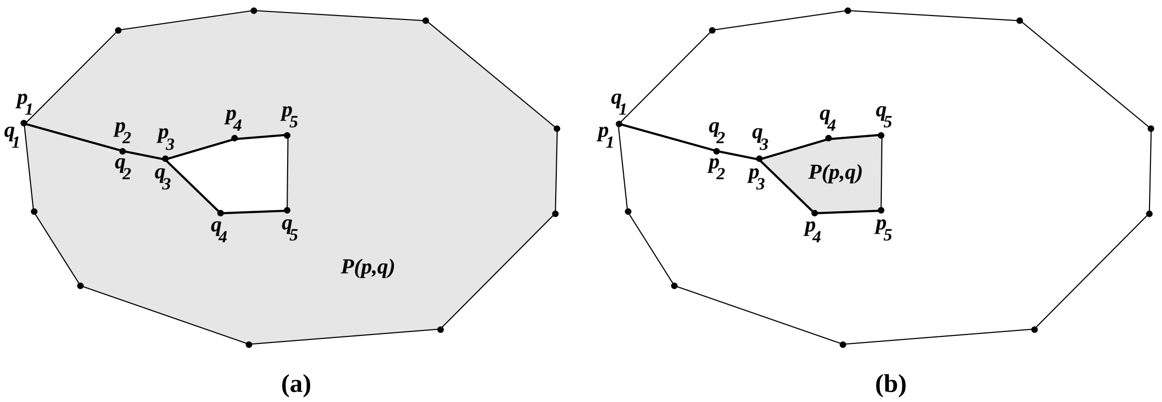

Figure 4: Legal polygons that are not simple polygons.

4(a)). The quadrangulation of $P(p,q)$ is defined to be the quadrangulation of P'. If $P(p,q)$ has no interior then its quadrangulation is null.

2.2 Partitioning of a Legal Polygon

The optimal partitioning of a legal polygon depends on the following two properties. The first is the cardinality of the legal polygon $P(p,q)$, which we denote as $P(p,q)\#$. By the cardinality of the legal polygon, we mean the number of triangles that a triangulation of the legal polygon would admit. It can be shown that $P(p,q)\# = (v-2)+2i$, where v is the number of vertices on $\delta P'$ and i is the number of vertices in the interior of P'. The second property is the parity of the boundary of the legal polygon, which we denote as $\mathcal{P}(\delta P(p,q))$. The parity of the boundary, $\mathcal{P}(\delta P(p,q))$, is determined by whether there are an even or odd number of vertices on $\delta P'$. The importance of the parity of the boundary of a legal polygon is explained by the following lemma.

Lemma 2.1 *Let P be a simple polygon. Also, let S be a given point set in the plane, including the vertices of P. If P admits a quadrangulation, then δP must have an even number of vertices.*

Proof. Same type of argument as in the proof for Lemma 3.1 in [2]. □

Lemma 2.2 *For any convex quadrangulation with respect to a point set, each interior vertex y is incident on an edge (x,y) that is l.w.r. y.*

Proof. Simple extension of Anagnostou and Corneil [1]'s proof for triangulations based on the fact that in a convex quadrangulation there are no interior angles greater than 180 degrees. □

A consequence of this property in lemma 2.2 is the following lemma.

Lemma 2.3 *Given a convex quadrangulation of a legal polygon $P(p,q)$ and a point $y \in S \cap P(p,q)$, there exists at least one legal path t using the edges of the quadrangulation with end point y such that t is completely within $P(p,q)$.*

Proof. From lemma 2.2, we know there exists a point x such that the edge (x,y) is l.w.r. to y. Since (x,y) must be an edge from the quadrangulation of $P(p,q)$, x is either in the interior of $P(p,q)$, or on $\delta P(p,q)$. If x is in the interior, then we continue as we did for y with another edge that is l.w.r. to x, forming a legal path until we either reach a point on the boundary of $CH(S)$ or a point on the boundary of $P(p,q)$. In the latter case, suppose, without loss of generality, that the point is on the path p. Then we can continue the legal path up to the first point of p, again reaching the boundary of $CH(S)$. $\square$

Definition 2.11 *Given a legal polygon $P(p,q)$, let u and v be the end points of legal paths p and q, respectively. Let w be a point $\in S$ but $\notin P(p,q)$ such that $\triangle uwv$ is an empty triangle. Define a QT-partition of $P(p,q)$ with respect to (w.r.t.) w to be an optimal partition of $P(p,q)$ into convex quadrangles and an empty triangle $\triangle uvz$ such that the quadrangle $\square uzvw$ is convex.*

An example of a QT-partition of $P(p,q)$ w.r.t. a vertex w is shown in figure 5. By lemma 2.1, a quadrangulation of $P(p,q)$ is only defined for a legal polygon $P(p,q)$ where $\mathcal{P}(\delta P(p,q)) =$ "even", whereas a QT-partition of $P(p,q)$ w.r.t. w is only defined for a legal polygon $P(p,q)$ where $\mathcal{P}(\delta P(p,q)) =$ "odd".

Definition 2.12 *Define $d(x,y)$ to be the weight of a line segment (x,y). For the MWCQ algorithm, this weight will be the Euclidean distance between x and y.*

Definition 2.13 *Define $W(p,q)$ to be the weight of a MWCQ of the legal polygon $P(p,q)$ where $W(p,q) = 2\times$(weights of all interior edges) + (weights of edges on $\delta P'$). If $P(p,q)$ does not admit*

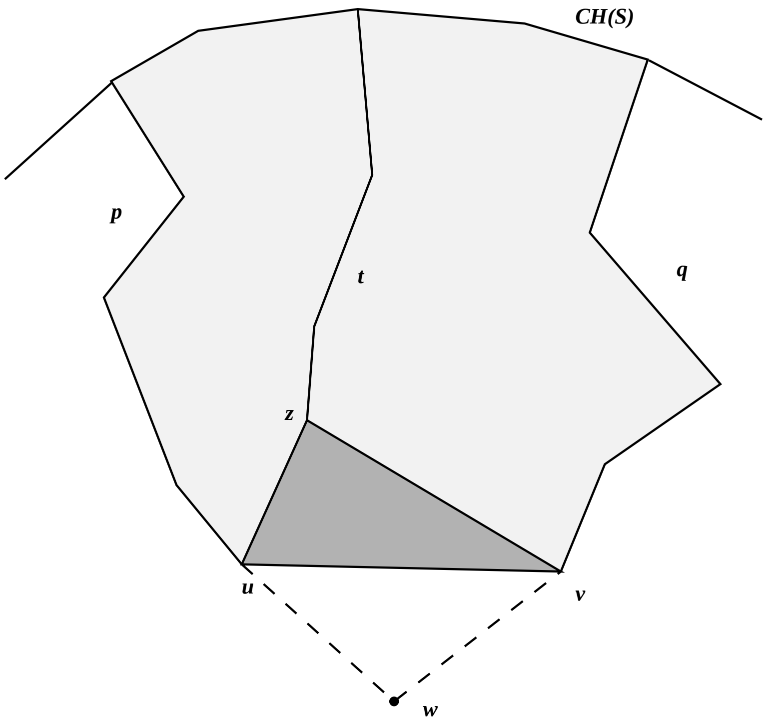

Figure 5: Definition of a QT-partition of $P(p,q)$ w.r.t. w.

a MWCQ, then $W(p,q) = \infty$. Also, define $OPT(p,q,w)$ be the weight of a QT-partition of $P(p,q)$ w.r.t. w where $OPT(p,q,w) = 2\times$(weights of all interior edges) + weights of edges on $\delta P' - d(u,v)$. Again, if $P(p,q)$ does not admit a QT-partition of $P(p,q)$ w.r.t. w, then $OPT(p,q,w) = \infty$.

The following data structure is used to simplify some of the algorithmic details.

Definition 2.14 *Consider three points $x, y, z \in S$. Define a point w to be strongly visible with respect to a line segment (x,y), if x, y, and w form an empty triangle. Let the data structure MVE(x,y) be the list of points from the point set ordered clockwise around x that are strongly visible from (x,y) and to the right of a directed line from x to y. MVE stands for "mutually visible and empty".*

Note that the clockwise ordering of strongly visible vertices to the right of (x,y) around x is the same as the clockwise ordering around y.

The minimum weight of a QT-partition of $P(p,q)$ w.r.t. w, $OPT(p,q,w)$, may be determined as in the following lemma.

Lemma 2.4 *Consider a legal polygon $P(p,q)$ with $\mathcal{P}(\delta P(p,q))$ = "odd", where u and v are the end points of the legal paths p and q, respectively. Let t be any legal path that is compatible with both p and q and is completely within $P(p,q)$. Let t have end point $z \in S \cap P(p,q)$ where $z \in MVE(v,u)$. Therefore, if vertex $w \notin P(p,q)$ and $w \in MVE(u,v)$, $OPT(p,q,w)$ may be determined by*

$$\begin{aligned} OPT(p,q,w) &= \min_{t,z \ni \Box_{uwvz} \text{ is convex}} (W(p,t) + W(t,q) + d(u,z) + d(z,v)) \\ &= \min_{z \ni \Box_{uwvz} \text{ is convex}} \left(d(u,z) + d(z,v) + \min_{t} (W(p,t) + W(t,q)) \right) \end{aligned}$$

Proof. By lemma 2.3, there exists at least one legal path t ending at each vertex $z \in MVE(v,u)$ where $z \in S \cap P(p,q)$. Each legal path t denotes a QT-partition of $P(p,q)$ with respect to vertex w such that $\Box_{uwvz}$ is convex. Therefore, it is possible to explore all possible QT-partitions of $P(p,q)$ to choose the optimal QT-partition for a given w. $\Box$

Note in the above lemma that since $W(p,t)$ and $W(t,q)$ are the weights of quadrangulations of $P(p,t)$ and $P(t,q)$, respectively, then by lemma 2.1, $\mathcal{P}(\delta P(p,t)) = \mathcal{P}(\delta P(t,q))$ = "even".

A straight forward determination of a MWCQ of a legal polygon would be as follows. Consider a legal polygon $P(p,q)$ with $\mathcal{P}(\delta P(p,q))$ = "even", where u and v are the end points of the legal paths p and q, respectively. Consider all pairs of vertices $z, w \in S \cap P(p,q)$ such that the quadrangle $\Box_{uwvz}$ is convex and empty. By lemma 2.3, there exists at least one legal path t completely within $P(p,q)$ ending at z and at least one legal path s completely within $P(t,q)$ ending at w, as exemplified in figure 6. By searching all possible choices of t and s such that p is compatible with t, t is

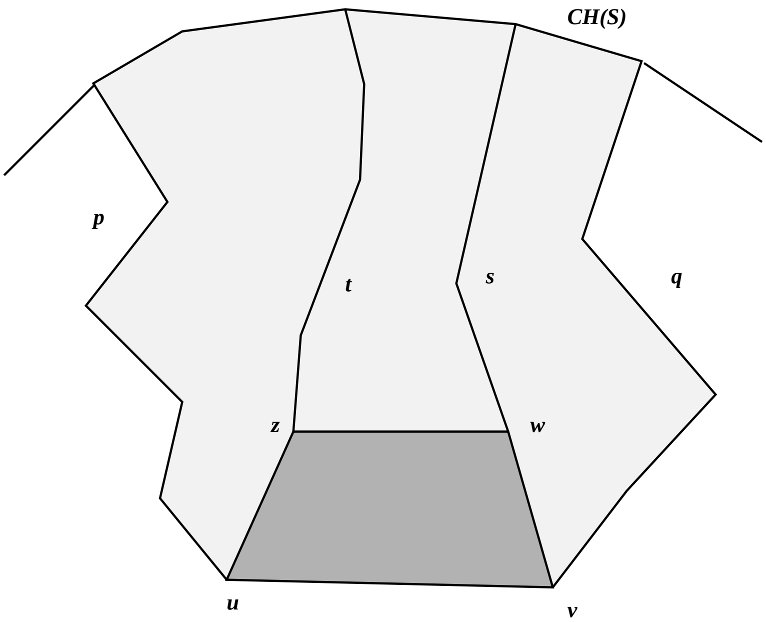

Figure 6: A brute force approach to find the MWCQ of $P(p,q)$.

compatible with s, and s is compatible with q, we can find a MWCQ of $P(p,q)$:

$$W(p,q) = \min_{t,z,s,w \ni \Box_{uwvz} \text{ is convex}} \begin{pmatrix} W(p,t) + W(t,s) + W(s,q) + \\ d(z,u) + d(u,v) + d(z,w) + d(w,v) \end{pmatrix}. \tag{1}$$

Again, note that since $W(p,t)$, $W(t,s)$, and $W(s,q)$ are the weights of quadrangulations of $P(p,t)$, $P(t,q)$ and $W(s,q)$, respectively, then by lemma 2.1, $\mathcal{P}(\delta P(p,t)) = \mathcal{P}(\delta P(t,s)) = \mathcal{P}(\delta P(s,q)) =$ "even". An algorithm based on equation (1) would have time complexity of $O(n^{4\hbar+2})$.

By using the $OPT(p,q,w)$ data structure, we find an alternate, more efficient, determination of a MWCQ of a legal polygon. This approach is very similar to that used by Conn and O'Rourke for the MWCQ problem for simple polygons [4] [5].

Lemma 2.5 *Given a legal polygon $P(p,q)$ with $\mathcal{P}(\delta P(p,q))$ = "even", consider all vertices $z \in S \cap P(p,q)$ such that $z \in MVE(v,u)$. Let t be any legal path completely within $P(p,q)$ ending at a* *given z. Therefore,*

$$W(p,q) = \min_{t,z} \left(W(p,t) + OPT(t,q,u) + d(z,u) + d(u,v) \right)$$

Proof. Again, consider all pairs of vertices $z, w \in S \cap P(p,q)$ such that quadrangle $\square_{uwvz}$ is convex and empty. Let t be any legal path completely within $P(p,q)$ ending at z and let s be any legal path completely within $P(p,q)$ ending at w such that p is compatible with t, t is compatible with s, and s is compatible with q. Therefore, equation 1 may be rewritten as follows.

$$\begin{aligned}
W(p,q) &= \min_{t,z,s,w \ni \square_{uwvz} \text{ is convex}} \begin{pmatrix} W(p,t) + W(t,s) + W(s,q) + \\ d(z,u) + d(u,v) + d(z,w) + d(w,v) \end{pmatrix} \\
&= \min_{t,z} \begin{pmatrix} W(p,t) + d(z,u) + d(u,v) + \\ \min_{s,w \ni \square_{uwvz} \text{ is convex}} \left(W(t,s) + W(s,q) + d(z,w) + d(w,v) \right) \end{pmatrix} \\
&= \min_{t,z} \left(W(p,t) + d(z,u) + d(u,v) + OPT(t,q,u) \right)
\end{aligned}$$

Since, by lemma 2.4, $OPT(t,q,u)$ is the minimum weight of a QT-partition of $P(t,q)$ w.r.t. u, and all possible choices of t and z are explored, including the choice of the optimal partition by lemma 2.3, then the minimum weight for $W(p,q)$ is found. $\square$

This last expression is shown diagrammatically in figure 7. Thus, at each stage of the algorithm, a legal polygon of $P(p,q)\# = m$ with $\mathcal{P}(\delta P(p,q)) =$ "even" can be partitioned into an empty quadrangle and three legal subpolygons (two of which are contained in the QT-partition of $P(t,q)$ w.r.t. u) of size smaller than m whose MWCQ had been determined at earlier stages of the algorithm.

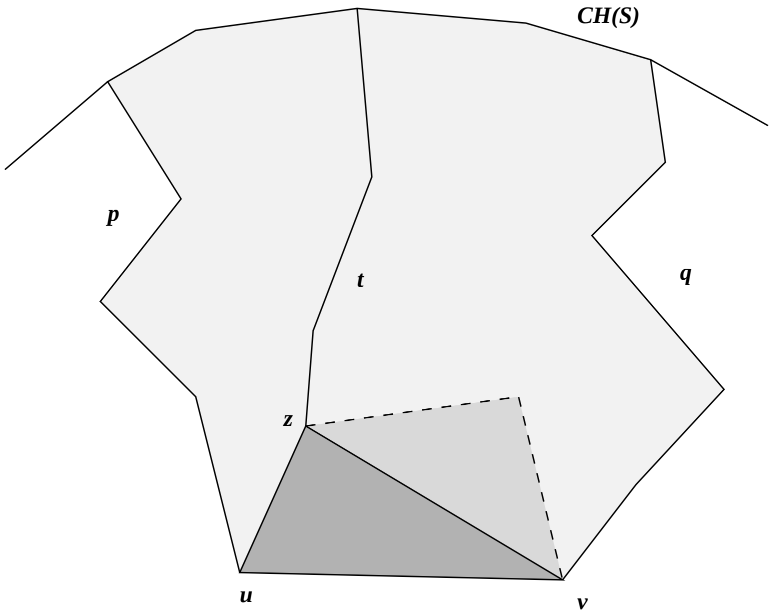

Figure 7: Finding a MWCQ for $P(p,q)$ using $OPT(t,q,u)$.

2.3 Implementation

We will define the following as an implementation detail which will help lower the overall time complexity of the algorithm.

Definition 2.15 *Consider a legal polygon $P(p,q)$, where u and v are the end points of the legal paths p and q, respectively. Let t be any legal path, with end point $z \in S \cap P(p,q)$, that is compatible with both p and q and is completely within $P(p,q)$. Define $\hat{t}$ to be the choice of t such that $W(p,t)+W(t,q)$ is minimised over all possible t for a given z. That is,*

$$\min_t \left(W(p,t)+W(t,q)\right) = W(p,\hat{t}) + W(\hat{t},q)$$

Therefore,

$$OPT(p,q,w) = \min_{z \ni \square_{uwvz} \text{ is convex}} \Big(d(u,z) + d(z,v) + W(p,\hat{t}) + W(\hat{t},q) \Big)$$

The MWCQ algorithm is as follows.

Algorithm A: Minimum Weight Convex Quadrangulation of a Point Set

Input: A point set S.

Output: The weight of the MWCQ is stored in $W(s,r)$ where r and s are two legal paths composed of single vertices that are clockwise consecutive on the boundary of $CH(S)$.

begin

for each polygon $P(p,q)$ such that $P(p,q)\# = 0$ **do**

$W(p,q) = 0$

end for

for i:= 1 to $(n-2)$ **do** /$\star$ $n-2$ is maximum number of triangles. $\star$/

for each polygon $P(p,q)$ such that $P(p,q)\# = i$ **do**

u := the end point of p; v := the end point of q

if $\mathcal{P}(\delta P(p,q))$ = "odd" **then**

1. For each $z \in MVE(v,u)$, compute $\hat{t}$
2. For each $w \in MVE(u,v)$ compute $OPT(p,q,w)$

else /$\star$ $\mathcal{P}(\delta P(p,q))$ = "even" $\star$/

3. $W(p,q) = \min_{t,z} (W(p,t) + OPT(t,q,u) + d(z,u) + d(u,v))$

end

Using a parallel data structure to $W(p,q)$ would also allow us to return the actual quadrangulation without affecting the overall complexity.

2.4 Correctness Analysis

Theorem 2.1 *Given a point set S, Algorithm A will find, if one exists, a MWCQ with respect to S.*

Proof. The proof will be done by induction on the cardinality of the legal polygons, $P(p,q)\#$. As before, u is the end point of p, and v is the end point of q. Basis cases: $P(p,q)\# \leq 2$. For $P(p,q)\# = 0$, $W(p,q) = 0$. For $P(p,q)\# = 1$, the interior of the legal polygon is an empty triangle. Let x be the third vertex of the triangle. Therefore, for each $w \in MVE(u,v)$ such that the quadrangle $\Box uxvw$ is convex, set $OPT(p,q,w) = d(u,x) + d(x,v)$. For $P(p,q)\# = 2$, the quadrangulation is optimal if the interior of the legal polygon is an empty convex quadrangle. In this case, $W(p,q)$ is given the sum of the weights of the edges of the quadrangle.

Induction hypothesis: Suppose that we have optimally partitioned all legal polygons $P(p,q)$ with $P(p,q)\# \leq k$, where k is a positive integer such that $k \geq 2$.

Induction step: Let $P(p,q)$ be a legal polygon with cardinality $k+1$ and let Q be an optimal partition of $P(p,q)$. Again, we have the two cases where $\delta P(p,q)$ has either an even or odd number of vertices. First, assume $\mathcal{P}(\delta P(p,q)) =$ "odd". Then the correctness of the value of $OPT(p,q)$ with respect to each $w \in MVE(u,v)$ is established by lemma 2.4. Next, assume $\mathcal{P}(\delta P(p,q)) =$ "even". Now, the correctness of value of $W(p,q)$ is established by lemma 2.5. By the inductive assumption, all legal polygons with cardinality $k+1$ are optimally partitioned. Let s and t be two consecutive paths on the boundary of $CH(S)$ that are composed of single vertices. Therefore, the algorithm finds the MWCQ of S, which is represented by $P(t,s)$ with $\mathcal{P}(\delta P(t,s)) =$ "even". $\square$

2.5 Complexity Analysis

Now, let us determine the overall time complexity of Algorithm A. First, consider the number of legal paths. Recall that all legal paths start from $CH_1(S)$. From each interior convex hull, $CH_i(S)$, $i > 1$, we can choose zero or one from $|CH_i(S)|$ possible points. Thus the total number of legal paths is bounded by

$$|CH_1(S)|(|CH_2(S)|+1)(|CH_3(S)|+1)\cdots(|CH_\hbar(S)|+1) = O(n^\hbar).$$

Therefore, the total number of legal polygons is $O(n^{2\hbar})$.

The complexity of the number of legal polygons is also the same complexity involved in generating them in the preprocessing step. Also, $MVE(x,y)$ may be initialised for all x,$y \in S$ in $O(n^3)$ time using a simple line sweep algorithm (see [15]).

In Algorithm A, the time complexity is dominated by the nested **for** loops which iterate $O(n^{2\hbar})$ times. The determination of $\hat{t}$ at the line labelled "1" takes $O(n^{\hbar+1})$ time. The initialisation of $OPT(p,q,w)$ over all w for a given p and q at the line labelled "2" can be shown to take an amortised $O(n \log n)$ time. For the line labelled "3", we search over $O(n)$ z and all possible associated legal paths t ending at z. In the worst case, as most $O(n^\hbar)$ legal paths t may end at z. Therefore, line "3" takes $O(n^{\hbar+1})$ time.

Therefore the overall time complexity is

$$O\left(n^{2\hbar}\right)+O\left(n^{2\hbar}(n^{\hbar+1}+n\log n+n^{\hbar+1})\right)=O(n^{3\hbar+1}),\hbar>1.$$

When $\hbar$ is a constant, this is a polynomial time complexity. For $\hbar > 1$, this complexity is the same as for the Minimum Weight Triangulation algorithm of Anagnostou and Corneil [1]. The space complexity is $O(n^{2\hbar+1})$ to store the $OPT(p,q,w)$ data structure, the largest data structure.

3 Conclusion

In this paper, we have presented an algorithm to find a MWCQ of a point set on the boundaries of nested convex hull boundaries. The time complexity is of the same order as the Minimum Weight Triangulation algorithm of Anagnostou and Corneil [1]. If a non-infinite weight is returned by Algorithm A, then it also returns the answer of affirmative whether a convex quadrangulation is admitted by the point set, as stated in the following theorem.

Theorem 3.1 *Given a point set S with $\hbar$ = constant, then Algorithm A will determine in polynomial time if S admits a convex quadrangulation.*

Proof. The point set S admits a convex quadrangulation if Algorithm A terminates with $W(t,s) \neq \infty$. The correctness and complexity of Algorithm A are discussed above. □

In the above development, we assumed that the points in the point set S were in general position. Although we have not explored the issue fully, we anticipate that removing this general position condition will require only a simple extension of our algorithm.

References

[1] Efthymios Anagnostou and Derek Corneil. Polynomial-time instances of the Minimum Weight Triangulation problem. *Computational Geometry: Theory and Applications*, 3:247–259, 1993.

[2] Prosenjit Bose and Godfried Toussaint. No quadrangulation is extremely odd. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput. (ISAAC 95)*, volume 1004 of *Lecture Notes in Computer Science*, pages 372–381. Springer-Verlag, 1995.

[3] J. Bown. *Injection Mouldings of Plastic Components.* McGraw-Hill, New York, 1979.

[4] H. E. Conn. *Some Polygon Decomposition Problems.* Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1990.

[5] H. E. Conn and J. O'Rourke. Minimum weight quadrilaterization in $O(n^3 \log n)$ time. In *Proc. 28th Allerton Conf. Commun. Control Comput.*, pages 788–797, October 1990.

[6] Charles A. Hall and Thomas A. Porsching. *Numerical Analysis of Partial Differential Equations.* Prentice Hall, 1990.

[7] E. Heighway. A mesh generator for automatically subdividing irregular polygons in to quadrilaterals. *IEEE Transactions on Magnetics*, 19(6):2535–2538, 1983.

[8] J. Kahn, M. Klawe, and D. Kleitman. Traditional galleries require fewer watchmen. *SIAM J. Algebraic Discrete Methods*, 4:194–206, 1983.

[9] J. M. Keil and J.-R. Sack. Minimum decompositions of polygonal objects. In G. T. Toussaint, editor, *Computational Geometry*, pages 197–216. North-Holland, Amsterdam, Netherlands, 1985.

[10] M. J. Lai. Scattered data interpolation and approximation by using C^1 piecewise cubic polynomials, submitted for publication.

[11] M. J. Lai and L. L. Schumaker. Scattered data interpolation using C^2 piecewise cubic polynomials of degree six. In *Third Workshop on Proximity Graphs*, Mississippi State University, Starkville, Mississippi, December 1 – 3, 1994.

[12] A. Lubiw. Decomposing polygonal regions into convex quadrilaterals. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 97–106, 1985.

[13] Henk Meijer and David Rappaport. Computing the minimum weight triangulation of a set of linearly ordered points. *Information Processing Letters*, 42:35–38, 1992.

[14] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[15] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, NY, 1985.

[16] J.-R. Sack. An $O(n \log n)$ algorithm for decomposing rectilinear polygons into convex quadrilaterals. In *Proc. 20th Allerton Conf. Commun. Control Comput.*, pages 64–75, 1982.

[17] J.-R. Sack and G. T. Toussaint. A linear-time algorithm for decomposing rectilinear polygons into convex quadrilaterals. In *Proc. 19th Allerton Conf. Commun. Control Comput.*, pages 21–30, 1981.

[18] J.-R. Sack and G. T. Toussaint. Guard placement in rectilinear polygons. In G. T. Toussaint, editor, *Computational Morphology: a Computational Geometric Approach to the Analysis of Form*, pages 153–175. North-Holland, Amsterdam, Netherlands, 1988.

[19] Granville Sewell. *The Numerical Solution of Ordinary and Partial Differential Equations.* Academic Press, 1988.

[20] Godfried Toussaint. Quadrangulations of planar sets. In S. Akl, F. Dehne, J-R Sack, and N. Santoro, editors, *Algorithms and Data Structures (Proc. WADS '95)*, volume 955 of *Lecture Notes in Computer Science*, pages 218–227, 1985.