Minimum Convex Partition 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*}*@cs.queensu.ca*

August 30, 1999

Abstract: *A convex partition 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 a convex polygon. A minimum convex partition with respect to S is a convex partition of S such that the number of convex polygons is minimised. In this paper, we will present a polynomial time algorithm to find a minimum convex partition with respect to a point set S where S is constrained to lie on the boundaries of a fixed number of nested convex hulls.*

1 Introduction

A convex partition (convex decomposition) 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 a convex polygon. A minimum convex partition (MCP) with respect to S is a convex partition of S such that the number of convex polygons is minimised.

With regards to polygonal domains, the first algorithm for finding an MCP for a simple polygon was due to Greene, whose algorithm for a simple polygon with n vertices and N reflex vertices (where the interior angle is reflex) runs in $O(N^2n^2)$ time [4]. Independently, Keil presented a dynamic programming algorithm that found an MCP of a simple polygon in $O(N^2 n \log n)$ time [5]. Recently, Keil and Snoeyink have shown that an MCP of a simple polygon can be found in $O(n+N^2\min\{N^2, n\})$ time [6]. It is also known that for polygons with polygonal holes, the problem is NP-hard [5].

With respect to point sets, there is no published work that we are aware of for the MCP problem. In this paper, we find an MCP with respect to a given point set S using the dynamic programming technique of "legal polygons" used by Anagnostou and Corneil for the Minimum Weight Triangulation problem for point sets [1]. The time complexity of the algorithm is polynomial in the cardinality of S, where S is constrained to lie on the boundaries of a fixed number of nested convex hulls. This work is an extension of work presented in a previous paper on the minimum weight convex quadrangulation problem for a constrained point set [3]. As mentioned, we use dynamic programming as in [1] and [3]. However, in both prior results, the fact that the cells in the decomposition have a constant number of sides (i.e., triangles, quadrangles) is implicit in the complexity results. A naive application of these prior methods would lead to an algorithm with a time complexity that is exponential in $O(n)$ due to the fact that the boundaries of the cells may now be of size $O(n)$ (this point is made in the comment that follows Definition 2.15). Thus a substantial new technique is needed for this extension of the previous results.

2 MCP of a Constrained Point Set

The approach of Anagnostou and Corneil is centered on a dynamic programming algorithm based on finding optimal partitions for a polynomial number of "legal polygons" of varying sizes (cardinality) defined from the point set, S [1]. At each stage of the algorithm, the optimal partition for a legal polygon is constructed from the best of all possible subdivisions of the polygon into partitioned legal subpolygons of smaller cardinality. Only one optimal solution is then kept for each legal polygon thus keeping the time complexity proportional to the number of legal polygons.

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 [7].

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 [8].

Let E^2 denote the Euclidean space in two dimensions. We assume that our 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.

Below are 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 the 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 segment (s, x) does not intersect $CH_{h(x)}(S)$.*

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

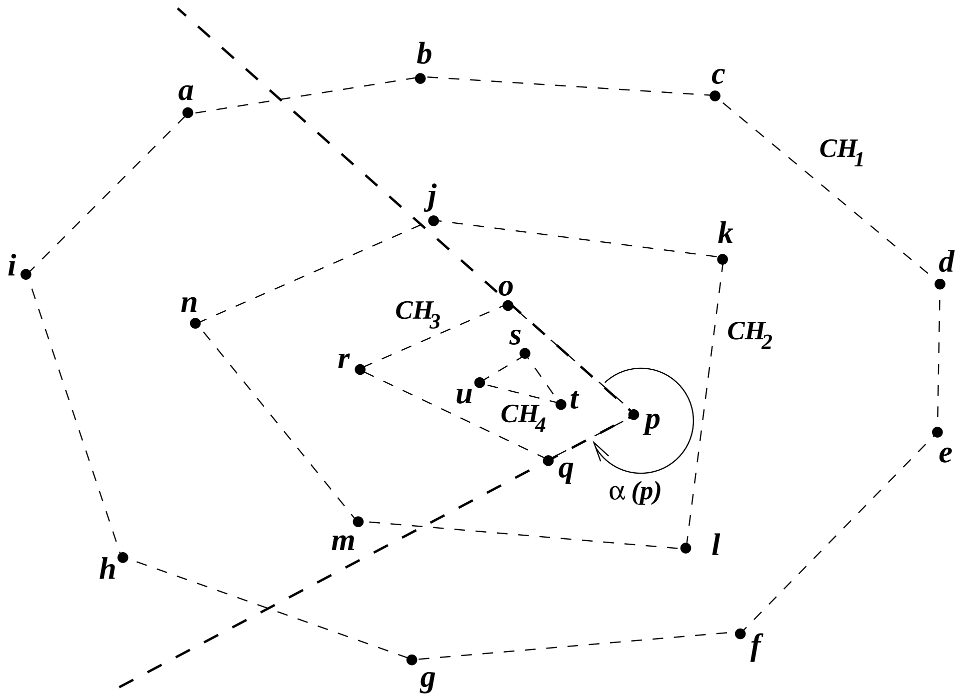

Figure 1: Nested convex hulls and related definitions. Adapted from [1].

For example, see figure 1. In this example, $\hbar = 4$, and the different convex hulls are labelled. For the vertex 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 partition.

Definition 2.4 *A path $(p_1, p_2, ..., p_m)$ is a simple planar straight-line graph (see [9]) with vertex 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 point, y, by a line segment (x, y) that is l.w.r. to y.*

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

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, i.e., $p_k \neq q_l$;

(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,

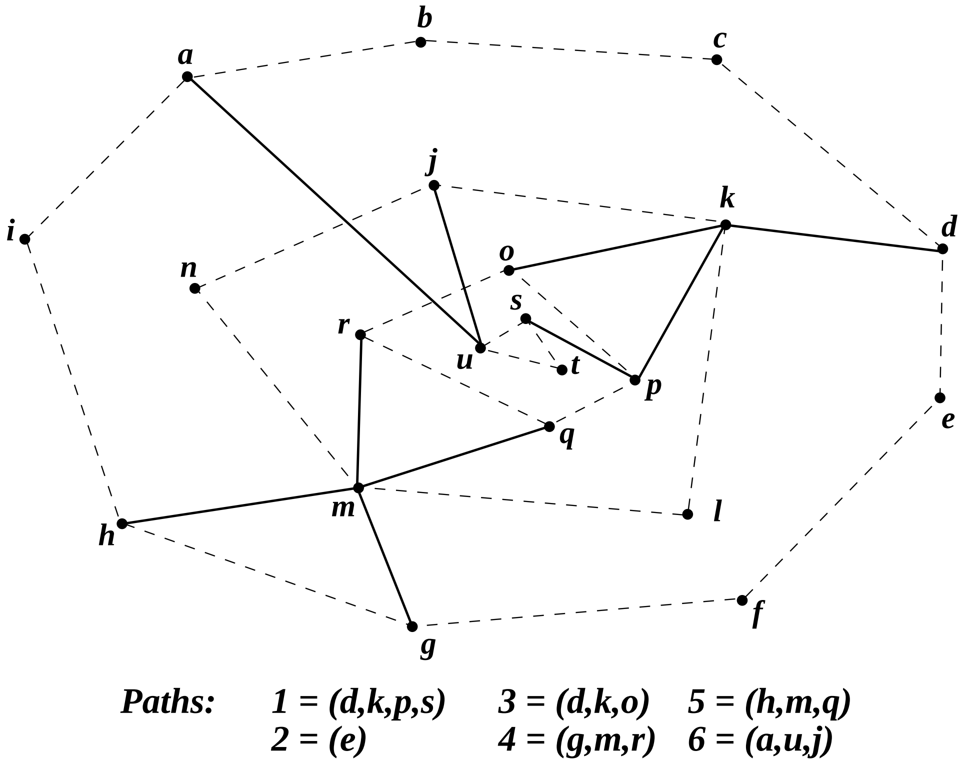

Figure 2: Examples of Paths. Adapted from [1].

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;

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

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 convex partitions 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 convex partition of P is a planar subdivision of P whose vertices are $S \cap P$ and where every face is a convex polygon.*

Definition 2.8 *Let P be a simple polygon and S be a point set in the plane. We say that P is empty if $(P \setminus \delta P) \cap S = \emptyset$.*

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

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

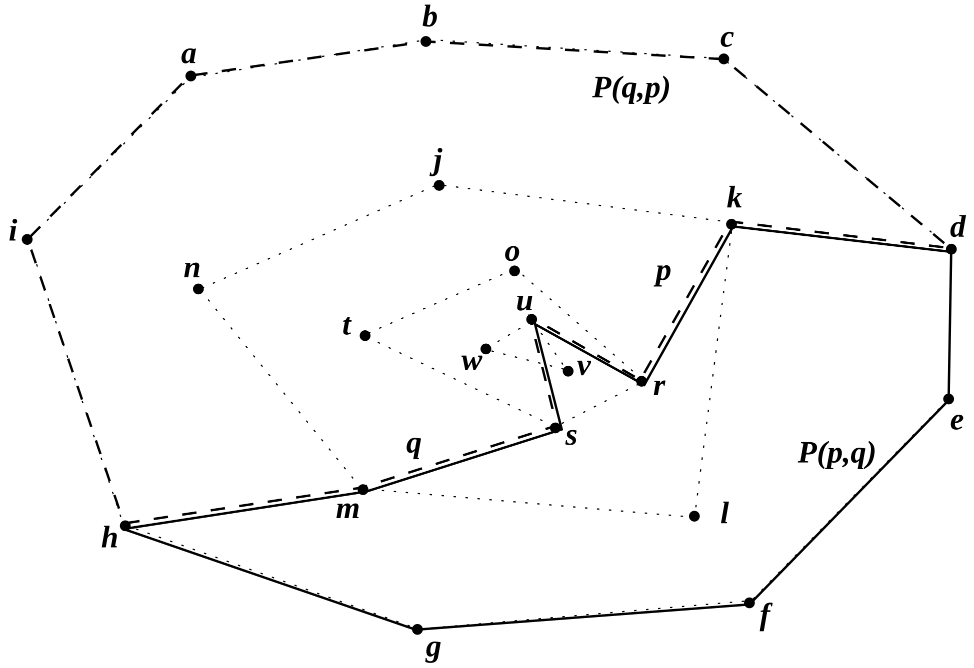

Figure 3: Example legal polygons with legal paths $p = (d, k, r, u)$ and $q = (h, m, s)$. Adapted from [1].

Definition 2.9 *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* $P(p, q)$ *and its boundary* $\delta P(p, q)$ *as follows. The boundary is*

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

The legal polygon $P(p, q)$ *is the closed region of the plane bounded by* $\delta P(p, q)$.

See figure 3 for examples for legal polygons. The legal polygons $P(p, q)$ and $P(q, p)$ use $p = (d, k, r, u)$ and $q = (h, m, s)$.

Definition 2.10 *Consider a legal polygon* $P(p, q)$ *and a legal path* t *with its endpoint* $z \in S \cap P(p, q)$. *We consider* t *to be "completely within* $P(p, q)$*" if* $t \subseteq P(p, q)$.

Note that if legal path s is completely within $P(p, q)$, and p and q share a vertex a, and s_1 and p_1 are the vertices of s and p on $CH(S)$, then the path from s_1 to a is identical to the path from p_1 to a. Also, a legal polygon is not strictly a simple polygon since some edges of p and q may be identical, as shown in figure 4. Therefore, the definition of a convex partition of a polygon does not strictly apply to legal polygons.

Definition 2.11 *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*

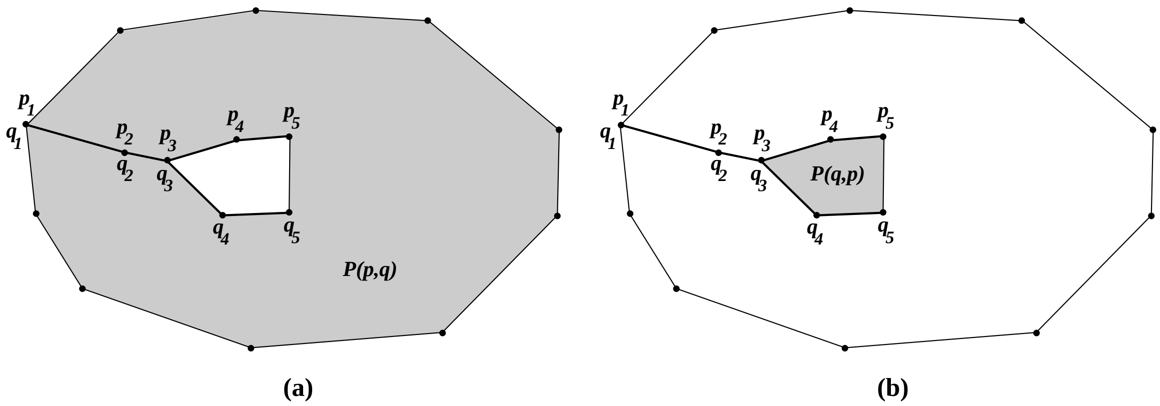

Figure 4: Legal polygons that are not simple polygons.

treat it as a simple polygon by considering any common edges of p *and* q *as being distinct (see figure 4(a)). The partition of* $P(p,q)$ *is defined to be the partition of* P'*. If* $P(p,q)$ *has no interior, its partition is null.*

2.2 Partitioning of a Legal Polygon

Let $P(p,q)\#$, the cardinality of $P(p,q)$, denote the number of triangles in a convex partition of $P(p,q)$ into triangles. The ordering mechanism used for our dynamic programming algorithm depends on $P(p,q)\#$. 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 points in the interior of P'.

Lemma 2.1 *For any convex partition with respect to a point set,* S*, each point* y *not on the boundary of CH*(S) *is incident on an edge* (x,y) *that is l.w.r. to* y*.*

Proof. Consider a convex partition of S, and consider any $y \in S$ interior to $CH(S)$ and the edges incident to y. Since the angles between these edges are less than 180 degrees, there must be at least one edge incident to y and exterior to $CH_{h(y)}(S)$. This edge is l.w.r. to y. □

A consequence of the property described in lemma 2.1 is the following lemma.

Lemma 2.2 *Given a convex partition 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 partition with end point* y *such that* t *is completely within* $P(p,q)$*.*

Proof. From lemma 2.1, 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 partition 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 former case, the legal path is complete. 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 $CH(S)$. □

The following data structure will be defined to simplify some of the algorithmic details.

Definition 2.12 *Consider two points $x, y \in S$. 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 counter-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 counter-clockwise ordering of strongly visible vertices to the right of (x, y) around x is the same as the counter-clockwise ordering around y.

Definition 2.13 *Let $[abc...d]$ denote a simple polygon P with vertices $a, b, c, ..., d$ in order along the boundary of P. Let the function $CVX(abc...d)$ return 1 if polygon $[abc...d]$ is convex and empty, 0 otherwise.*

Definition 2.14 *Define a k-gon to be a simple polygon with k vertices defining its boundary. Let k_m be the largest value of k where S admits an empty convex k-gon.*

Definition 2.15 *Define $N(p, q)$ to be the number of convex polygons in an MCP of the legal polygon $P(p, q)$.*

A straightforward determination of an MCP of a legal polygon would be as follows. Consider a legal polygon $P(p, q)$, where u and v are the endpoints of the legal paths p and q, respectively. Consider all choices of points $z_i \in P(p, q)$ such that the 3-gon $[uz_1v]$ is empty, or the 4-gon $[uz_1z_2v]$ is convex and empty, etc., up to the k_m-gon $[uz_1z_2...z_{k_m-2}v]$ being convex and empty. Also, we only consider such $k-$gons that are within $P(p, q)$. By lemma 2.2, there exists at least one legal path t_i completely within $P(p, q)$ ending at each z_i. By searching all possible combinations of the choices of $t_1, t_2, ..., t_i$, $1 \le i \le k_m - 2$, with $t_0 = p$ and $t_{i+1} = q$, such that p is compatible with t_1, and t_j is compatible with t_{j+1} and completely within $P(t_{j-1}, t_{j+1})$ where $1 \le j \le i$, we can find an MCP of $P(p, q)$:

$$N(p,q) = \min_{i=1,2,\ldots,k_m-2} \left\{ \min_{\substack{t_1,t_2,\ldots,t_i \ni CVX(uz_1z_2\ldots z_iv)=1 \\ \textit{and } [uz_1z_2\ldots z_iv] \subseteq P(p,q)}} \left\{ \begin{array}{l} N(p,t_1) + N(t_1,t_2) + \cdots \\ +N(t_{i-1},t_i) + N(t_i,q) \end{array} \right\} \right\} + 1 \quad (1)$$

This recurrence relation leads to an exponential solution for finding an MCP with respect to the point set S. Therefore, we need to consider an alternate approach for determining an MCP with respect to S.

Definition 2.16 *Given a legal polygon $P(p, q)$, let u and v be the end points of legal paths p and q, respectively. Let $w_1, w_2, ..., w_l$ be a subsequence of $MVE(u, v)$ such that $w_i \in P(q, p)$ for $i = 1..l$. Define a "k-partition of $P(p, q)$ with respect to (w.r.t.) $w_1, w_2, ..., w_l$" to be a partition of $P(p, q)$ into an empty k-polygon $[uz_1z_2...z_{k-2}v]$ at the (u, v) face such that the $(k + l)$-polygon $[w_lw_{l-1}...w_2w_1uz_1z_2...z_{k-2}v]$ is convex and empty, and the rest of $P(p, q)$ is partitioned into the minimum number of convex polygons.*

An example of a k-partition of $P(p, q)$ w.r.t. vertices $w_1, w_2, ..., w_l$ is shown in figure 5.

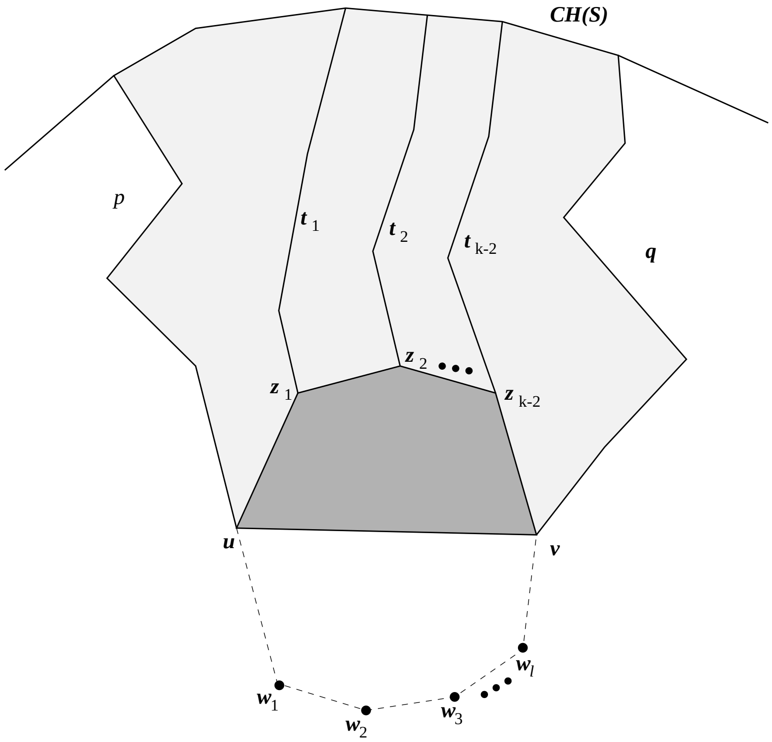

Figure 5: Definition of a k-partition of $P(p,q)$ w.r.t. $w_1, w_2, ..., w_l$.

Definition 2.17 *If $2 \leq k < k_m$ and $1 \leq l \leq k_m - k$, define $\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k)$ to be the minimum number of convex polygons minus one in a k-partition of $P(p,q)$ w.r.t. $w_1, w_2, ..., w_l$. If no such k-partition is admitted by $P(p,q)$, then $\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k)$ is set to $+\infty$. Also, if $k = 2$, then $\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, 2) = N(p,q)$.*

The value of $\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k)$ may be determined as follows.

Lemma 2.3 *Given k, $3 \leq k \leq k_m - l$, consider a legal polygon $P(p,q)$, where u and v are the end points of the legal paths p and q, respectively. In the following minimisation, t, with end point z, refers to all possible legal paths that are compatible with p and q. If $w_1, w_2, ..., w_l$ are vertices $\in P(q,p)$, $\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k)$ may be determined by*

$$\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k) = \min_{t \ni [uzv] \subseteq P(p,q)} \{N(p,t) + \mathcal{O}_{l+1}(t, q, u, w_1, w_2, ..., w_l, k-1)\} \quad (2)$$

Proof. Let $t_i, 1 \leq i \leq k-2$ be any legal paths, with end points $z_i \in P(p,q)$, that are completely within $P(p,q)$ such that p is compatible with t_1, t_j is compatible with t_{j+1} and completely within $P(t_{j-1}, t_{j+1})$ where $1 \leq j < i$, and t_i is compatible with q and completely within $P(t_{i-1}, q)$. Then we have

$$\begin{aligned}&\mathcal{O}_l(p, q, w_1, w_2, ..., w_l, k) \\ &\quad = \min_{\substack{t_1, t_2, ..., t_{k-2} \ni CVX(w_l w_{l-1} ... w_2 w_1 u z_1 z_2 ... z_{k-2} v) = 1 \\ and\ [u z_1 z_2 ... z_{k-2} v] \subseteq P(p,q)}} \{N(p,t_1) + N(t_1,t_2) + ... + N(t_{k-2}, q)\}\end{aligned}$$

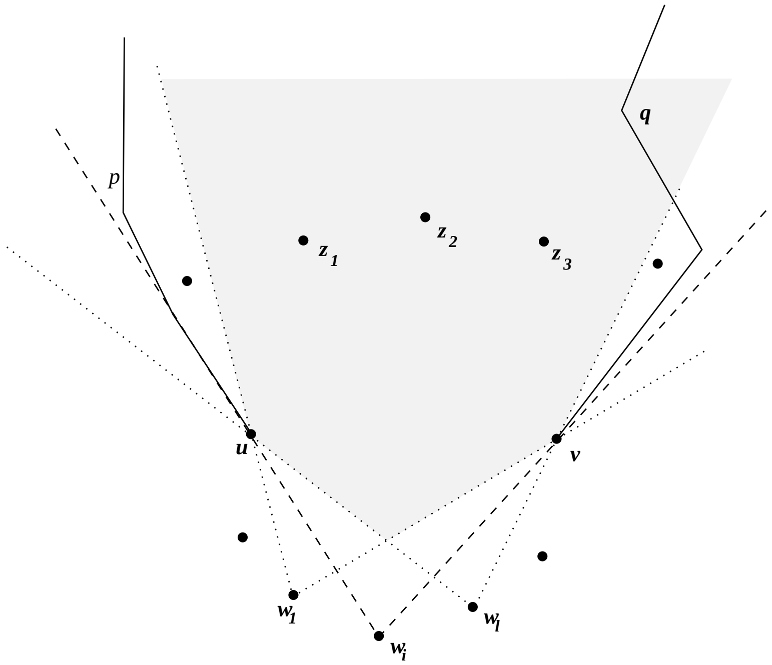

Figure 6: Diagram for Proof for lemma 2.4.

$$= \min_{t_1 \ni [uz_1v] \subseteq P(p,q)} \left\{ N(p,t_1) + \min_{\substack{t_2,\ldots,t_{k-2} \ni CVX(w_l w_{l-1} \ldots w_2 w_1 u z_1 z_2 \ldots z_{k-2} v)=1 \\ and\ [z_1 z_2 \ldots z_{k-2} v] \subseteq P(p,q)}} \{N(t_1,t_2) + \ldots + N(t_{k-2},q)\} \right\}$$

$$= \min_{t_1 \ni [uz_1v] \subseteq P(p,q)} \{N(p,t_1) + \mathcal{O}_{l+1}(t_1, q, u, w_1, w_2, \ldots, w_l, k-1)\}$$

□

The following lemma shows that we only need to consider $\mathcal{O}_l()$ for $l \leq 2$.

Lemma 2.4 *Consider a legal polygon $P(p,q)$. If $CVX(uvw_lw_{l-1}\ldots w_2w_1) = 1$ and $l \geq 2$ then*

$$\mathcal{O}_l(p, q, w_1, w_2, \ldots, w_l, k) = \mathcal{O}_2(p, q, w_1, w_l, k).$$

Proof. Suppose $\mathcal{O}_2(p, q, w_1, w_l, k)$ finds a $k-$partition of $P(p,q)$ w.r.t. w_1, w_l with the convex k-gon $[u, z_1, z_2, \ldots, z_{k-2}, v]$ as part of the partition at the face (u,v) at the endpoints of p and q, respectively. Also, suppose $\mathcal{O}_l(p, q, w_1, w_2, \ldots, w_l, k)$, for some value of l, $1 \leq l \leq k_m - k - 2$, finds a k-partition of $P(p,q)$ w.r.t. $w_1, w_2, \ldots, w_l$ with the convex k-gon $[u, z'_1, z'_2, \ldots, z'_{k-2}, v]$ as part of the partition at the face (u,v). We will consider a vertex y to be visible to a set of vertices $V = \{v_1, v_2, \ldots, v_k\}$, if for any vertex $x \in V$, $(x,y) \subseteq$ the polygon $[y, v_1, v_2, \ldots, v_k]$. See figure 6. In this diagram for the lemma, it is obvious that the vertices z_i that are visible to vertices $w_1, w_2, \ldots, w_{l-1}$, and w_l are determined by what is visible to the end points w_1 and w_l (the shaded area on the diagram). The position of any of the $w_i \in \mathrm{MVE}(u,v)$ between w_1 and w_l will not decrease the area visible to the end points w_1 and w_l unless $[uw_1w_iw_lv]$ has a reflex vertex. Therefore, an optimal $k-$partition that determines the value of $\mathcal{O}_{l+2}(p, q, w_1, w_2, \ldots, w_{l-1}, w_l, k)$ is also a valid $k-$partition of $P(p,q)$ w.r.t. w_1, w_l, and vice-versa. Therefore, $\mathcal{O}_{l+2}(p, q, w_1, w_2, \ldots, w_{l-1}, w_l, k) = \mathcal{O}_2(p, q, w_1, w_l, k)$. □

Motivated by lemma 2.4, the determination of $N(p,q)$ for each legal polygon $P(p,q)$ is done as in the following lemma.

Lemma 2.5 *Given a legal polygon $P(p,q)$, let u and v be the end points of legal paths p and q. In the following minimisations, t, with end point z, refers to all possible legal paths that are compatible with p and q such that $[uzv]$ is an empty 3-gon within $P(p,q)$.*

a) For each k, $3 \le k \le k_m - 2$, and for each pair of vertices $a, b \in P(q,p)$ such that $[auvb]$ is an empty convex 4-gon,

$$\mathcal{O}_2(p,q,a,b,k) \quad = \min_{t \ni z \text{ “sees” } a,b} \{N(p,t) + \mathcal{O}_2(t,q,u,b,k-1)\} \tag{3}$$

b) For each k, $3 \le k \le k_m - 1$, and for each vertex $a \in P(q,p)$ such that $[auv]$ is an empty 3-gon,

$$\mathcal{O}_1(p,q,a,k) \quad = \min_{t \ni z \text{ “sees” } a} \{N(p,t) + \mathcal{O}_2(t,q,u,a,k-1)\} \tag{4}$$

c) Finally,

$$N(p,q) = 1 + \min_{i=1,..,k_m-2} \left\{ \min_t \{N(p,t) + \mathcal{O}_1(t,q,u,i+1)\} \right\} \tag{5}$$

Proof. The proof for recurrence relation (3) follows from lemmas 2.3 and 2.4. And the proof for recurrence relation (4) follows directly from lemma 2.3. The proof for recurrence relation (5) is as follows. From equation (1), we have

$$\begin{aligned}
& \min_{t_1,t_2,\ldots,t_i \ni CVX(uz_1z_2\ldots z_iv)=1} \{N(p,t_1) + N(t_1,t_2) + \cdots + N(t_{i-1},t_i) + N(t_i,q)\} \\
= \; & \min_{t_1} \left\{ N(p,t_1) + \min_{t_2,\ldots,t_i \ni CVX(uz_1z_2\ldots z_iv)=1} \{N(t_1,t_2) + \ldots + N(t_i,q)\} \right\} \\
= \; & \min_{t_1} \{N(p,t_1) + \mathcal{O}_1(t_1,q,u,i+1)\}
\end{aligned}$$

□

In our algorithm, we apply lemma 2.5 to each $P(p,q)$ in increasing order of cardinality. When the algorithm is done, the solution is stored in $N(a,b)$ where a and b are two single vertex paths from two consecutive vertices on $CH(S)$. Using parallel data structures to $N(p,q)$, $\mathcal{O}_2(p,q,a,b,k)$, and $\mathcal{O}_1(p,q,a,k)$ would also allow us to return the actual convex partition without affecting the overall complexity.

2.3 Analysis of the Algorithm

2.3.1 Correctness Analysis

Theorem 2.1 *Given a point set S, an algorithm based on lemma 2.5 will find an MCP with respect to S.*

Proof. The proof will be done by induction on the cardinality of the legal polygons, $P(p,q)\#$. The correctness of the algorithm follows from lemma 2.5 with the following base cases, $P(p,q)\# = 0$. For each cases, $N(p,q) = 0$ and for $k > 2, \mathcal{O}_2(p,q,a,b,k) = \mathcal{O}_1(p,q,a,k) = +\infty \; \forall a,b$. □

2.3.2 Complexity Analysis

Now, let us determine the overall time complexity of our algorithm. 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. The value of k_m can be determined in $O(n^3)$ time [2].

In the algorithm itself, the same work is done for each legal polygon, $P(p,q)$, as per lemma 2.5.

1. The determination of $\mathcal{O}_2(p,q,a,b,k)$ takes $O(k_m(n^2)(n^\hbar))$ time for $O(n^2)$ choices of a and b. The minimisation over t requires looking $O(n^\hbar)$ choices of t;
2. The determination of $\mathcal{O}_1(p,q,a,k)$ takes $O(k_m(n)(n^\hbar))$ time for $O(n)$ choices of a. Again, the minimisation over t requires looking at $O(n^\hbar)$ choices of t;
3. The determination of $N(p,q)$ takes $O(k_m(n^\hbar))$ time where the minimisation over t requires looking at $O(n^\hbar)$ choices of t.

Since the determination of $\mathcal{O}_2(p,q,a,b,k)$ dominates the time complexity, this gives an overall time complexity of $O(k_m n^{3\hbar+2})$. And, since $k_m \leq n$, this algorithm has a time complexity of $O(n^{3\hbar+3})$. The space complexity is $O(n^{2\hbar+3})$ to store the $\mathcal{O}_2(p,q,a,b,k)$ data structure, the largest data structure.

Observe that our dynamic programming algorithm can optimise other functions. For example, instead of minimising the number of convex cells in the decomposition, we can obtain a minimum weight convex decomposition by minimising the overall sum of the weights of the edges.

Acknowledgements

The authors wish to thank the anonymous referees for their suggestions and corrections to this paper.

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] D. Avis and D. Rappaport. Computing the largest empty convex subset of a set of points. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 161–167, 1985.

[3] T. Fevens, H. Meijer, and D. Rappaport. Minimum weight convex quadrangulation of a constrained point set. In *Abstracts from 2th Annual CGC Workshop on Computational Geometry*, Durham, NC, October 1997. (Accepted for publication in International Journal of Computational Geometry and Applications, 1999).

[4] D. H. Greene. The decomposition of polygons into convex parts. In F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Adv. Comput. Res.*, pages 235–259. JAI Press, London, England, 1983.

[5] J. M. Keil. Decomposing a polygon into simpler components. *SIAM J. Comput.*, 14:799–817, 1985.

[6] M. Keil and J. Snoeyink. On the time bound for convex decomposition of simple polygons. In *Proceedings of the 10rd Canadian Conference on Computational Geometry*, pages 54–55, 1998.

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

[8] J. O'Rourke. *Computational Geometry in C (Second Edition).* Cambridge University Press, 1998.

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