Finding a Hausdorff Core of a Polygon: On Convex Polygon Containment with Bounded Hausdorff Distance*

Reza Dorrigiv[1], Stephane Durocher[2,1], Arash Farzan[1], Robert Fraser[1], Alejandro López-Ortiz[1] **, J. Ian Munro[1], Alejandro Salinger[1], and Matthew Skala[1,3]

[1] Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada, {rdorrigiv,sdurocher,afarzan,r3fraser,alopez-o, imunro,ajsalinger,mskala}@cs.uwaterloo.ca

[2] Department of Computer Science, University of Manitoba, Winnipeg, Canada, durocher@cs.umanitoba.ca

[3] Department of Computer Science, University of Toronto, Toronto, Canada, mskala@ansuz.sooke.bc.ca

Abstract. Given a simple polygon P, we consider the problem of finding a convex polygon Q contained in P that minimizes $H(P, Q)$, where H denotes the Hausdorff distance. We call such a polygon Q a *Hausdorff core* of P. We describe polynomial-time approximations for both the minimization and decision versions of the Hausdorff core problem, and we provide an argument supporting the hardness of the problem.

1 Introduction

Traditional hierarchical representations allow for efficient storage, search and representation of spatial data. These representations typically divide the search space into areas for which membership can be tested efficiently. If the query region does not intersect a given area, the query can proceed without further consideration of that area. When a space or object has certain structural properties, the data structure built upon it can benefit from those properties. For example, the data structure of Kirkpatrick [12] is designed to index planar subdivisions answering point queries in time $O(\log n)$ and space $O(n)$, with preprocessing time $O(n \log n)$.

Our study is motivated by the problem of path planning in the context of navigation at sea. In this application, a plotted course must be tested against bathymetric soundings to ensure that the ship will not run aground. We suppose the soundings have been interpolated into contour lines [1] and the plotted course is given as a polygonal line. There is no requirement of monotonicity or even

* Funding for this research was made possible by the NSERC strategic grant on Optimal Data Structures for Organization and Retrieval of Spatial Data.

** Part of this work took place while the fifth author was on sabbatical at the Max-Planck-Institut für Informatik in Saarbrücken, Germany.

continuity between contour lines in the map. A given line might be a maximum, minimum or a falling slope. Similarly, we observe that in general there are several disconnected contour lines with the same integer label (depth).

Although contour lines can be arbitrarily complicated, typical shipping routes run far from potential obstacles for the majority of their trajectories, and only short segments require more careful route planning. As a result, most intersection checks should be easy: we should be able to subdivide the map into areas such that most of our intersection tests are against conveniently-shaped areas, reserving more expensive tests for the rare cases where the path comes close to intersecting the terrain.

The search for easily-testable areas motivates the study of the simplification of a contour line into a simpler object which is either entirely contained within the contour line or fully contains it. In this paper we consider the case in which the simplified polygon must be convex and contained.

1.1 Definitions

A *polygon* P is a closed region in the plane bounded by a finite sequence of line segments or *edges*. We restrict our attention to *simple polygons*, in which the intersection of any two edges is either empty or an endpoint of each edge and the intersection of any three edges is empty. Finally, recall that a region P is *convex* if for all points p and q in P, the line segment $\overline{pq}$ is contained in P.

Given a simple polygon P and a metric d (defined on polygons), a *d-core* of P is a convex polygon Q contained in P that minimizes $d(P, Q)$. Examples of metrics d of interest include the area of the region $P \setminus Q$, the Hausdorff distance between P and Q, and the link distance (which is a discrete distance metric). A common measure of distance between two sets P and Q is given by

$$d(P, Q) = \max \left\{ \max_{p \in P} \min_{q \in Q} \operatorname{dist}(p, q), \max_{q \in Q} \min_{p \in P} \operatorname{dist}(p, q) \right\}.$$

When P and Q are polygons in the plane and $\operatorname{dist}(p, q)$ denotes the Euclidean (ℓ_2) distance between points p and q, $d(P, Q)$ corresponds to the *Hausdorff distance* between sets P and Q, which we denote by $H(P, Q)$. We define the corresponding d-core as the *Hausdorff core*. We consider both the minimization and decision versions of problem of finding a Hausdorff core for a given simple polygon P:

Input. A simple polygon P.
Question. Find a Hausdorff core of P.

Input. A simple polygon P and a non-negative integer k.
Question. Does there exist a convex polygon Q contained in P such that $H(P, Q) \leq k$?

The *1-centre* of a polygon P (also known as Euclidean centre) is the point c that minimizes the maximum distance from c to any point in P. In this work we are only interested in the 1-centre inside P, also known as constrained Euclidean centre. Although the unconstrained 1-centre is unique, this is not necessarily true for the constrained version [6]. A constrained 1-centre of a polygon P of n vertices can be computed in time $O(n \log n + k)$, where k is the number of intersections between P and the furthest point Voronoi diagram of the vertices of P [6]. For simple polygons $k \in O(n^2)$. Note that the constrained 1-centre of P is a point $c \in P$ that minimizes $H(P, c)$. Throughout the rest of the paper, when we refer to a 1-centre, we specifically mean a constrained 1-centre.

1.2 Related Work

We can divide the problem of approximating polygons into two broad classes: inclusion problems seek an approximation contained in the original polygon, while enclosure problems determine approximation that contains the original polygon. Formally, let $\mathcal{P}$ and $\mathcal{Q}$ be classes of polygons and let μ be a function on polygons such that for polygons P and Q, $P \subseteq Q \Rightarrow \mu(P) \leq \mu(Q)$. Chang and Yap [7] define the inclusion and enclosure problems as:

- $Inc(\mathcal{P}, \mathcal{Q}, \mu)$: Given $P \in \mathcal{P}$, find $Q \in \mathcal{Q}$ included in P, maximizing $\mu(Q)$.
- $Enc(\mathcal{P}, \mathcal{Q}, \mu)$: Given $P \in \mathcal{P}$, find $Q \in \mathcal{Q}$ enclosing P, minimizing $\mu(Q)$.

The best known enclosure problem is the convex hull, which we may state formally as $Enc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$, where $\mathcal{P}_{\text{simple}}$ is the family of simple polygons and $\mathcal{P}_{\text{con}}$ is the family of convex polygons. Given a convex polygon P, many problems are tractable in linear time: $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_3, \text{area})$ [16], $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_3, \text{perimeter})$ [5], and $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_{\text{par}}, \text{area})$ [17], where $\mathcal{P}_{\text{par}}$ is the family of parallelograms. For general k-gons, $Enc(\mathcal{P}_{con}, \mathcal{P}_k, \text{area})$ can be solved in $O(kn + n \log n)$ time [3].

Perhaps the best known inclusion problem is the potato-peeling problem of Chang and Yap [7], defined as $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$. There is an $O(n^7)$ time algorithm for this problem, and an $O(n^6)$ time algorithm when the measure is the perimeter, $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{perimeter})$, where n is the number of vertices of P [7]. The problem of finding the triangle of maximal area included in a convex polygon, $Inc(\mathcal{P}_{con}, \mathcal{P}_3, \text{area})$, can be solved in linear time [9]. The generalization of this problem to any k-gon can be solved in time $O(kn + n \log n)$ [2]. If the input polygon is not restricted to be convex, $Inc(\mathcal{P}_{con}, \mathcal{P}_3, \text{area})$ can be found in time $O(n^4)$ [15].

The inclusion and enclosure problems can also be formulated as minimizing or maximizing a measure $d(P, Q)$. Note that in the case when $\mu(Q)$ is the area, maximizing or minimizing $\mu(Q)$ for the inclusion and enclosure problems, respectively, is equivalent to minimizing the difference in areas ($d(P, Q) = |\mu(P) - \mu(Q)|$). Both the inclusion and enclosure problems using the Hausdorff distance as a measure were studied by Lopez and Reisner [14], who present polynomial-time algorithms to approximate a convex polygon minimizing the Hausdorff distance to within an arbitrary factor of the optimal. Since the input polygon is convex, the approximating solution is restricted to a maximum number of vertices. In the same work, the authors also studied the *min-#* version of the problem, where the goal is to minimize the number of vertices of the approximating polygon,

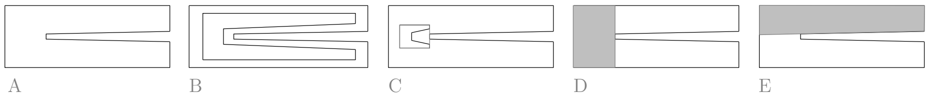

Fig. 1. A. The input polygon P. **B.** "Shrinking" the polygon. **C.** Shrink until the convex hull is contained in P. **D.** The solution returned by the Chassery and Coeurjolly algorithm. **E.** An actual solution.

given a maximum allowed error. For this setting, they show that the inclusion and enclosure problems can be approximated to within one vertex of the optimal in $O(n \log n)$ time and $O(n)$ time, respectively.

The inclusion problem that minimizes the Hausdorff distance where the input is a simple (not necessarily convex) polygon was addressed in [8]. They present an algorithm that returns a Hausdorff core for the case when the point 1-centre is contained in the input polygon P. The algorithm shrinks the input polygon P until its convex hull is contained in the original P. If the shrunken polygon P' is not convex, the region in which the convex hull P' intersects P is removed from P'. The procedure is repeated starting with P' until a convex polygon is obtained. In general, the algorithm does not return a Hausdorff core if the point 1-centre is not contained in P. A counterexample is illustrated in Figure 1. To the best of our knowledge, no algorithm for finding a Hausdorff core of an arbitrary simple polygon, $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{Hausdorff})$, has appeared in the literature.

2 Preliminary Observations

In this section we make several observations about properties of polygons, convex polygons, and the Hausdorff distance in the context of the discussed problem. These observations will be useful in later sections in establishing our main results. Due to lack of space, we omit the proofs.

Given a polygon P and a convex polygon Q inside P, it suffices to optimize the maximum distance from points $p \in P$ to polygon Q to obtain a Q with a minimum Hausdorff distance:

Observation 1 *Given any simple polygon P and any convex polygon Q contained in P, $\max_{p \in P} \min_{q \in Q} d(p,q) \geq \max_{q \in Q} \min_{p \in P} d(p,q)$. Therefore,*

$$H(P,Q) = \max_{p \in P} \min_{q \in Q} d(p,q).$$

Among the points of P and Q, the Hausdorff distance is realized at the vertices of P. Furthermore, it occurs between Q and vertices that lie on the convex hull of P:

Lemma 1. *Given a simple polygon P and a convex polygon Q contained in P,*

$$H(P,Q) = H(CH(P)_V, Q),$$

where $CH(P)$ denotes the convex hull of set P and for any polygon A, A_V denotes the set of vertices of set A.

$H(P,Q)$ is determined by the vertices of P that lie on the convex hull of P, however all vertices and edges of P must be considered to determine whether Q is contained in P. The decision version of the Hausdorff core problem with parameter k is defined as follows; we consider circles of radius k centered at vertices $CH(P)_V$ and ask whether there exists a convex polygon Q such that it intersects all such circles:

Observation 2 *Let $C_k(p)$ denote a circle of radius k centred at p. Given a simple polygon P and a convex polygon Q contained in P,*

$$H(P,Q) \leq k \Leftrightarrow \forall p \in CH(P),\ C_k(p) \cap Q \neq \varnothing.$$

Finally, we wish to know some point contained in Q. If the 1-centre of P is not in Q, then Q intersects some vertex of P:

Lemma 2. *Given a simple polygon P and a convex polygon Q contained in P, let P_{1c} be the constrained 1-centre of P. At least one point in the set $\{P_{1c}, P_V\}$ is contained in Q if Q is a Hausdorff core of P. Let a point chosen arbitrarily from this set be Q_p.*

3 Hausdorff Core Minimization Problem

In this section we outline an algorithm to solve the Hausdorff core problem which operates by shrinking circles centred on selected vertices of P (which vertices have circles is discussed shortly). Invariant 1 must hold for a solution to exist:
Invariant 1. There exists a set of points $\{p_1, p_2, \ldots, p_k\}$, where k is the current number of circles, such that $\forall i\ p_i \in C_i$ and $\forall i, j, i \neq j\ \overline{p_i p_j}$ does not cross outside the original simple polygon.

Invariant 1 implies that a solution Q with $H(P,Q) = r$ exists, where r is the radius of the circles. We sketch the solution in Algorithm 1, and we illustrate an example of the operation of the algorithm in Figure 2. We find P_{1c} using the technique of [6]; there may be multiple such vertices, but we can choose one arbitrarily. A solution is not unique in general, but we find a polygon Q which minimizes $H(P,Q)$.

3.1 Proof of Correctness

The solution Q is a convex polygon that intersects every circle. If each circle C_i touches the solution convex polygon Q, we know that the distance from each vertex with a circle to Q is at most r, the radius of C_i. If a vertex $v \in CH(P)_V$ does not have a circle, then $\text{dist}(v, Q_p) \leq r$. Therefore, given a simple polygon P, this algorithm finds a convex polygon Q contained in P such that $\forall p \in CH(P)_V, \exists q \in Q$ s.t. $d(p,q) \leq r$. By Lemma 1, we know that Q is a solution where $H(P,Q) = r$. It remains to be shown that there does not exist a convex polygon Q' such that $\text{dist}(p, q') \leq r'$, where $r' < r$. This cannot be the case, for if the circles were shrunk any further, no convex polygon could intersect some

Algorithm 1 Hausdorff Core Minimization Algorithm

HCORE(P)
$Q = \emptyset, r_{\min} = \infty$
for each $Q_p \in \{P_{1c}, P_V\}$ **do**
Begin with circles of radius r_0 centred on the vertices $v \in CH(P)_V$, where $r_0 = \text{dist}(v_f, Q_p)$ and $v_f = \arg\max_{p \in P} \text{dist}(p, Q_p)$.
Any circle centred at a vertex v where $\text{dist}(Q_p, v) < r$ contains Q_p; such circles are ignored for now.
Reduce the radius such that at time $t_i \in [0, 1]$, each circle has radius $r(t_i) = r_0 \times (1 - t_i)$. Let $Q(t_i)$ be a solution at time t_i, if it exists. The radius is reduced until one of three events occurs:
(1) $r(t_i) = \text{dist}(Q_p, v_n)$, where v_n is the farthest vertex from Q_p that is not the centre of a circle. Add a circle centred at v_n with radius $r(t_i)$.
(2) $Q(t_i)$ cannot cover Q_p. In this case, we break and if $r(t_i) < r_{\min}$, then set $Q = Q(t_i)$ and $r_{\min} = r(t_i)$.
(3) A further reduction of r will prevent visibility in P between two circles. Again, we break and if $r(t_i) < r_{\min}$, then set $Q = Q(t_i)$ and $r_{\min} = r(t_i)$.
end for
return Q

pair of the circles by Invariant 1. Therefore, the polygon would necessarily be of distance $\text{dist}(p, q') > r'$ for some vertex p.

Finally, the optimality of the algorithm is guaranteed since we search different possibilities for the point Q_p which is contained in the solution Q. By Lemma 2, we know that at least one such point Q_p is contained in the optimal solution. By trying all possibilities, we ensure that the globally optimal solution is obtained.

4 Algorithmic Complexity of the Problem

The decision version of the exact problem consists of determining whether we can draw a polygon with one vertex in or on each circle and each successive pair of vertices is able to see each other around the obstructions formed by vertices of the input. For any fixed choice of the obstructing vertices, this consists of a system of quadratic constraints of the form "variable point in circle" and "two variable points collinear with one constant point." For the optimization version we need only make the circle radius a variable and minimize that. This is a simple mathematical programming problem, potentially tractable with a general solver.

Solving systems that include quadratic constraints is in general NP-hard; we can easily reduce from 0-1 programming by means of constraints of the form $x(x - 1) = 0$. Nonetheless, some kinds of quadratic constraints can be addressed by known efficient algorithms. Lobo et al. [13] describe many applications for second-order cone programming, a special case of semidefinite programming. The "point in circle" constraints of our problem can be easily expressed as second-order cone constraints, so we might hope that our problem could be expressed as a second-order cone program and solved by their efficient interior point method.

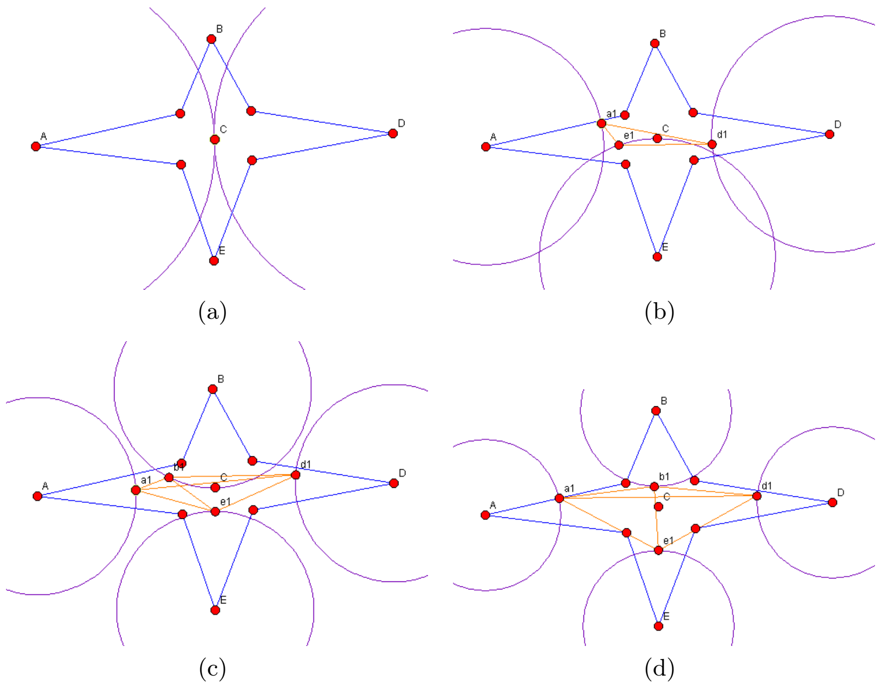

Fig. 2. (a) Two circles are centred on the critical points v_f. (b) $\text{dist}(E, 1c) = r$, so we add a new circle centred on E of radius r. The orange (light) lines indicate lines of visibility between the circles. (c) Another circle is added centred at point B. (d) We cannot shrink the circles any further, otherwise Invariant 1 would be violated. Therefore, a solution can be composed from the orange line segments.

However, the "two variable points collinear with one constant point" constraints are not so easy to handle. With (x_1, y_1) and (x_2, y_2) the variable points and (x_C, y_C) the constant point, we have the following:

$$\frac{y_1 - y_C}{x_1 - x_C} = \frac{y_2 - y_C}{x_2 - x_C} \tag{1}$$

$$x_2 y_1 - x_2 y_C - x_C y_1 = x_1 y_2 - x_1 y_C - x_C y_2 \tag{2}$$

This constraint is hyperbolic because of its cross-product terms. The techniques of Lobo et al. [13] can be applied to some hyperbolic constraints, subject to limitations whose basic purpose is to keep the optimization region convex.

As shown in Figure 3, it is possible for our problem to have two disconnected sets of solutions, even with as few as four circles. For a point A on the first circle, we can trace the polygon through the constant point B to that edge's intersection with the second circle at C, then through the constant point D and so on around to H. The lines AB and GH intersect at I, which is our choice for one vertex of the polygon, the others being C, E, and G. If I is inside the

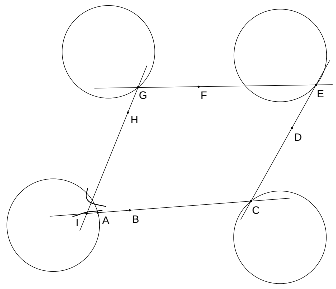

Fig. 3. Two disconnected solution intervals

circle, we have a feasible solution. But the heavy curves show the locus of I for different choices of A, and the portion of it inside the circle is in two disjoint pieces. The set of solutions to the problem as shown is disjoint, corresponding to a slice (for a constant value of the circle-radius variable) through a non-convex optimization region. As a result, neither second-order cone programming nor any other convex optimization technique is immediately applicable.

5 An Approximation Algorithm Hausdorff Core

5.1 The Decision Problem

First we discuss the decision version of the approximation algorithm, where were are given a distance r and wish to know whether there is an approximate Hausdorff core solution with $H(P,Q') \leq r + 2\varepsilon'$. This approximation scheme seeks to grow circles by an additive factor ε', and determine whether there exists a solution for these expanded circles. We still require that the approximate solution Q' must not cross outside P, and that Invariant 1 holds. Given ε as input, where ε is a small positive constant, we calculate $\varepsilon' = d_{vf} \cdot \varepsilon$ as the approximation factor of $H(P,Q)$. Recall that d_{vf} is the distance from the constrained 1-centre P_{1c} to the most distant vertex $v_f \in P$. Notice that this method of approximation maintains a scale invariant approximation factor, and the size of the of the approximation factor for a given P is constant, regardless of Q and the magnitude of r.

The strategy behind this approximation scheme is that by growing the circles by ε', they may be discretized. Consequently, it is possible to check for strong visibility between discrete intervals, which avoids some of the problems faced by the exact formulation of the problem. One of the challenges of this approach is the selection of the length of the intervals on the new circles of radius $r + \varepsilon'$. We require that the intervals be small enough so that we will find a solution for the approximation if one existed for the original circle radius. In other words, given an exact solution Q for the original radius r such that $H(P,Q) \leq r$, we are guaranteed that at least one interval on each of the expanded circles will be contained inside Q.

First we determine whether the polygon can be approximated by a single line segment. We construct an arc segment of radius $2d_{vf}$ (the maximum diameter of P) and arc length ε'. The interior angle of the circular segment C_φ formed by this arc is $\varphi = \varepsilon'/2d_{vf} = \varepsilon/2$. If an interior angle of Q' is less than or equal to φ, then Q' may be fully covered by C_φ since Q' is convex. In this case, there exists a line segment Q_ℓ which approximates Q' such that $H(Q', Q_\ell) < \varepsilon'$.

To determine whether Q can be approximated by a line segment, we grow all existing circles by a further factor of ε', so that they have radius $r^\star = r + 2\varepsilon'$. Since Q is convex, this operation means that a line segment which approximates Q will now intersect at least one arc from each circle. By Lemma 2, we know that $P_c \in \{P_{1c}, P_V\}$ is contained in Q. Therefore, we attempt to find a line intersecting a point P_c and a segment of each circle of radius $r^\star$ for each P_c. For a selected P_c, we build an interval graph in the range $[0...\pi]$. For each circle C_i, if a line at angle $\theta \mod \pi$ from an arbitrary reference line intersects a segment of C_i contained in P before intersecting P itself, then C_i covers θ in the interval graph. If there is a non-zero intersection between all circles in the interval graph, then the solution is a line segment Q_ℓ at angle θ to the reference line, intersecting P_c with endpoints at the last circles that Q_ℓ intersects. Therefore, if there exists a solution $H(P, Q) \leq r$ where Q can be approximated by a line segment Q_ℓ with $H(Q, Q_\ell) < 2\varepsilon'$, then we will find Q_ℓ.

If we have not found a solution Q_ℓ, we know that all interior angles of Q are greater than φ, and so we wish to determine an approximating polygon Q'. If we divide the expanded circle of radius $r + \varepsilon'$ into $12\pi/(\varepsilon^2 d_{vf})$ equal intervals, at least one would be fully contained in Q regardless of where the intervals are placed on the circle. Now finding Q' is simply a matter of finding a set of intervals such that there exists one interval on each circle which has strong visibility with an interval on all the other circles, and then selecting one point from each interval. A solution has the form $Q' = \{q_1 \dots q_k\}$, where q_i is a point on C_i in the interval contained in the solution.

We use a dynamic programming algorithm to find a solution given a set of circles in the input polygon. We use a table $A[i, j]$ that stores, for a pair of intervals i and j in different circles, a range of possible solutions that include those intervals (See Figure 4). We find the convex polygon that includes intervals i and j by combining two convex polygons, one that includes i and an interval k^* and another that includes j and k^*. In order to compute $A[i, j]$ we lookup the entries for $A[i, k_1] \dots A[i, k_m]$ and $A[k_1, j] \dots A[k_m, j]$, where $k_1, \dots, k_m$ are the intervals of a circle k, to determine if there is such k^* for which there are solutions $A[i, k^*]$ and $A[k^*, j]$ that can be combined into one convex polygon. There are many solutions that include a certain pair of intervals, but we store only $O(n)$ solutions for each pair. For example, for the entry $A[i, j]$ we would store the edge coming out of j that minimizes the angle Θ for each choice of an edge coming out of interval i, as shown in Figure 4. This would be done recursively at each level, which would make partial solutions easier to combine with other solutions while keeping convexity. Note that a particular choice of pairs of circles to form the solution Q' corresponds to a triangulation of Q', and since there are $O(n)$ pairs

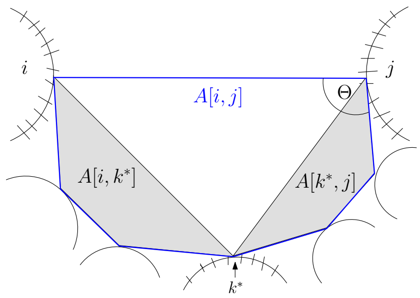

Fig. 4. The convex polygon that includes intervals i and j is built by combining a polygon that includes i and k^* and one that includes j and k^* (painted in grey).

of vertices joined in the triangulation, we need to store entries for the intervals of $O(n)$ pairs of circles. Given the clique of strongly visible intervals, we may now freely choose a point from each interval to obtain the solution polygon Q'. We run the dynamic programming algorithm iteratively for each $P_c \in \{P_{1c}, P_V\}$, using only circles centred on vertices $v \in P_V$ where $\text{dist}(v, P_c) < r$. If no solution Q' is found for any P_c, then there is no solution where $H(P,Q) = r$.

We present the following observations pertaining to Q and Q':

- $\exists Q \Rightarrow \exists Q'$, $\neg\exists Q' \Rightarrow \neg\exists Q$. The intervals are defined such that at least one interval from each circle will be contained in Q'.
- $\exists Q' \nRightarrow \exists Q$. The existence of Q' does not imply the existence of Q because the optimal solution may have circles of radius $r + \nu d_{vf}$, where $\nu < \varepsilon$.

5.2 The Minimization Problem

Given an optimal solution polygon Q where $H(P,Q) = r_{OPT}$, our algorithm finds an approximate solution Q' such that $H(P,Q') < r_{OPT} + 3\varepsilon'$. To determine a value of r' such that $r' \leq r_{OPT} + 3\varepsilon'$, it suffices to perform a binary search over possible values for r' in the range of $[0 \ldots v_f]$ executing the decision approximation algorithm at each iteration. At the i^{th} iteration of the algorithm, let the current radius be r_i. If the algorithm finds a solution Q_i such that $H(P,Q_i) = r_i$, we shrink the circles and use $r_{i+1} = r_i - dvf/2^i$. If the algorithm fails to find a solution, we use $r_{i+1} = r_i + dvf/2^i$. Initially, $r_0 = d_{vf}$, and the stopping condition is met when we find an approximate solution for radius r, and the approximate decision algorithm fails for radius $r - \varepsilon'$. Thus, the minimization version of the approximation algorithm requires $O(\log(\varepsilon^{-1}))$ iterations of the decision algorithm to find a solution. In the decision version, we showed that $H(Q,Q') < 2\varepsilon'$, if Q exists. In the minimization version, the best solution for a value of r may approach ε' less than the optimal value located on one of the radius intervals. Therefore, the minimization algorithm returns a solution Q' where $H(P,Q') < r_{OPT} + 3\varepsilon'$.

5.3 Running Time and Space Requirements

First we estimate the space and running time of the approximate decision algorithm. We compute the 1-centre using the technique in [6], which takes $O(n^2)$ time. The line solution tests a line against $O(n)$ circles, each of which may have $O(n)$ segments. This procedure is repeated $O(n)$ times, so this requires $O(n^3)$ time in total. In the dynamic programming table, there are $O(n)$ pairs of circles. The number of intervals on each circle is bounded by $O(\varepsilon^{-2})$, so we have $O(\varepsilon^{-4})$ possible combinations of intervals between two circles. Therefore there are $O(n\varepsilon^{-4})$ entries in the table, and each of them stores a description of $O(n)$ solutions. Hence the table needs roughly $O(n^2\varepsilon^{-4})$ space. If the number of entries in the table is $O(n\varepsilon^{-4})$, the dynamic programming algorithm should run in time $O(n\varepsilon^{-6})$, since in order to calculate each entry we need to check all the $O(\varepsilon^{-2})$ intervals of one circle. The algorithm may require $O(n)$ iterations to test each value of P_c, so the approximate decision algorithm requires $O(n^3 + n^2\varepsilon^{-6})$ time. Finally, the minimization version of the algorithm performs $O(\log(\varepsilon^{-1}))$ iterations of the approximate decision algorithm, so the complete algorithm requires $O((n^3 + n^2\varepsilon^{-6})\log(\varepsilon^{-1}))$ time to find an approximate solution.

6 Discussion and Directions for Future Research

The d-core problem is defined for any metric on polygons; we chose the Hausdorff metric, but many others exist. A natural extension of the Hausdorff metric might consider the *average* distance between two polygons instead of the *maximum*. This metric could be defined as follows:

$$H'(P,Q) = \max\left\{ \int_{p\in P} \min_{q\in Q} \operatorname{dist}(p,q)\ dp, \int_{q\in Q} \min_{p\in P} \operatorname{dist}(p,q)\ dq \right\},$$

where $\operatorname{dist}(p,q)$ denotes the Euclidean (ℓ_2) distance between points p and q. If Q is a point, then finding a point Q that minimizes $H'(P,Q)$ for a given polygon P corresponds to the continuous Weber problem, also known as the continuous 1-median problem. In the discrete setting, no algorithm is known for finding the exact position of the Weber point [4]. Furthermore, the problem is not known to be NP-hard nor polynomial-time solvable [11]. That suggests our problem may be equally poorly-behaved. Fekete et al. [10] considered the continuous Weber problem under the ℓ_1 distance metric.

In our original application, we hoped to create a hierarchy of simplified polygons, from full-resolution contour lines down to the simplest possible approximations. Then we could test paths against progressively more accurate, and more expensive, approximations until we got a definitive answer. We would hope to usually terminate in one of the cheaper levels. But our definition of d-core requires the core to be convex. Convexity has many useful consequences and so is of theoretical interest, but it represents a compromise to the original goal because it only provides one non-adjustable level of approximation. It would be interesting to consider other related problems that might provide more control over the approximation level.

Therefore, a direction for further work would be to define some other constraint to require of the simplified polygon. For instance, we could require that it be star-shaped, i.e. there is some point $p \in P$ such that every $q \in P$ can see p. A similar but even more general concept might be defined in terms of link distance.

Acknowledgements The authors would like to thank Diego Arroyuelo and Barbara Macdonald for their participation in early discussions of the problem, and the anonymous reviewers for their useful comments and suggestions.

References

1. P.K. Agarwal, L. Arge, T.M. Murali, K.R. Varadarajan, and J.S. Vitter. I/O-efficient algorithms for contour-line extraction and planar graph blocking. In *Proc. SODA*, pages 117–126. SIAM, 1998.
2. A. Aggarwal, M.M. Klawe, S. Moran, P. Shor, and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1):195–208, 1987.
3. A. Aggarwal and J. Park. Notes on searching in multidimensional monotone arrays. In *Proc. SFCS*, pages 497–512. IEEE, 1988.
4. C. Bajaj. The algebraic degree of geometric optimization problems. *Disc. & Comp. Geom.*, 3:177–191, 1988.
5. B.K. Bhattacharya and A. Mukhopadhyay. On the minimum perimeter triangle enclosing a convex polygon. In *Proc. JCDCG*, volume 2866 of *LNCS*, pages 84–96. Springer, 2002.
6. P. Bose and G. Toussaint. Computing the constrained Euclidean geodesic and link center of a simple polygon with applications. In *Proc. CGI*, page 102. IEEE, 1996.
7. J. S. Chang and C. K. Yap. A polynomial solution for the potato-peeling problem. *Disc. & Comp. Geom.*, 1(1):155–182, 1986.
8. J.-M. Chassery and D. Coeurjolly. Optimal shape and inclusion. In *Mathematical Morphology: 40 Years On*, volume 30, pages 229–248. Springer, 2005.
9. D.P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. In *Proc. SFCS*, pages 9–17, 1979.
10. S. P. Fekete, J. S. B. Mitchell, and K. Weinbrecht. On the continuous Fermat-Weber problem. *Oper. Res.*, 53:61–76, 2005.
11. S.L. Hakimi. Location theory. In Rosen, Michaels, Gross, Grossman, and Shier, editors, *Handbook Disc. & Comb. Math.* CRC Press, 2000.
12. D. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comp.*, 12(1):28–35, 1983.
13. M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Lin. Alg. & App.*, 284(1–3):193–228, 1998.
14. M.A. Lopez and S. Reisner. Hausdorff approximation of convex polygons. *Comp. Geom. Theory & App.*, 32(2):139–158, 2005.
15. E.A. Melissaratos and D.L. Souvaine. On solving geometric optimization problems using shortest paths. In *Proc. SoCG*, pages 350–359. ACM, 1990.
16. J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin. An optimal algorithm for finding minimal enclosing triangles. *J. Alg.*, 7:258–269, 1986.
17. C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, and B.L. Evans. Minimal enclosing parallelogram with application. In *Proc. SoCG*, pages 434–435. ACM, 1995.