Theory Comput. Systems **38**, 411–423 (2005)
DOI: 10.1007/s00224-004-1180-4

Theory of Computing Systems

Efficient Update Strategies for Geometric Computing with Uncertainty

Richard Bruce,[1] Michael Hoffmann,[1] Danny Krizanc,[2] and Rajeev Raman[1]

[1]Department of Mathematics and Computer Science, University of Leicester,
Leicester LE1 7RH, England
{r.bruce,m.hoffmann,r.raman}@mcs.le.ac.uk

[2]Department of Mathematics and Computer Science, Wesleyan University,
Middletown, CT 06459, USA
dkrizanc@wesleyan.edu

Abstract. We consider the problems of computing *maximal points* and the *convex hull* of a set of points in two dimensions, when the points are "in motion." We assume that the point locations (or trajectories) are *not* known precisely and determining these values exactly is feasible, but expensive. In our model the algorithm only knows areas within which each of the input points lie, and is required to identify the maximal points or points on the convex hull correctly by *updating* some points (i.e., determining their location exactly). We compare the number of points updated by the algorithm on a given instance to the minimum number of points that must be updated by a nondeterministic strategy in order to compute the answer provably correctly. We give algorithms for both of the above problems that always update at most three times as many points as the nondeterministic strategy, and show that this is the best possible. Our model is similar to that in [3] and [5].

1. Introduction

In many applications, an intrinsic property of the data one is dealing with is that it is "in motion," i.e., changing value (within prescribed limits) over time. For example, the data may be derived from a random process such as stock market quotes or queue lengths in switches or it may be positional data for moving objects such as planes in an air traffic control area or users in a mobile ad hoc network. Much work has gone into developing on-line algorithms [2] and kinetic data structures [1] in order to compute efficiently in these situations.

Most of the previous approaches to data in motion assume that the actual data values are known precisely at all times or that there is no cost in establishing these values. This is not always the case. In reality finding the exact value of some data item may involve costs in time, energy, money, bandwidth, etc. Accurate and timely stock quotes cost money. Remote access to the state of network queues costs time and bandwidth. Querying battery-powered units of sensor networks unnecessarily uses up precious energy.

In order to study the costs associated with updating data in an uncertain environment, we consider the *update complexity* of a problem. A problem instance consists of a function of n inputs to be computed (e.g., the maxima) and a specification of the possible values each of the inputs might obtain (e.g., a set of n real intervals). An update strategy is an adaptive algorithm for deciding which of the inputs should be updated (i.e., be determined exactly) in order to compute the function correctly. Consider a nondeterministic strategy that guesses a set of inputs to update and then verifies that the given set is sufficient to compute the function correctly. Let OPT be the size of the smallest guessed set for which the nondeterministic strategy is able to verify the correctness of the function value computed. An update strategy is said to *c-update optimal* if it updates at most $c * OPT + O(1)$ inputs. The notion of update complexity is implicit in Kahan's [5] model for data in motion. In the spirit of online competitive analysis he defined the *lucky* ratio of an update strategy on a sequence of queries whereby a strategy competes against a "lucky" (i.e., nondeterministic) strategy. Kahan provides update optimal strategies for the problems of finding the maximum, median and the minimum gap of a set of n real values constrained to fall in a given set of n real intervals.

Motivated by the situation where one maintains in a local cache, intervals containing the actual (changing) data values stored at a remote location, Olston and Widom [7] studied a similar notion. In their model the costs associated with updating data items may vary with the item and the function need not be computed exactly but only to within a given tolerance. A series of papers [3], [4], [6] establish tradeoffs between the update costs and the error tolerance and/or give complexity results for computing optimal strategies for such problems as selection, sum, average and computing shortest paths.

For the most part, the above results assume the uncertainty of a data item is best described by a real interval that contains it. In a number of situations, the uncertainty is more naturally captured by regions in two- (or higher) dimensional space. This is especially the case for positional data of moving objects with known upper bounds to their speed and possible constraints on their trajectories. The functions to be computed in these situations are most often geometric in nature. For example, to establish which planes to deal with first, an air traffic controller would be interested in computing the closest pair of points in three dimensions. To apply greedy directional routing in a mobile ad hoc network, a node must establish which of its neighbors is currently closest to the destination of a packet it is forwarding. To determine the extent of the coverage area of a mobile sensor network, one would like to compute the convex hull of the sensor's current positions.

In this paper we describe a general method, called the *witness algorithm*, for establishing upper bounds on the update complexity of geometric problems. The witness algorithm is used to derive update optimal strategies for the problems of finding all maximal points and reporting all points on the convex hull of a set of moving points whose uncertainty may be described by the closure of open connected regions on the plane. The restriction to connected regions is necessary in order to ensure the existence of strategies with bounded update complexity. For both of these problems we provide examples that show our update strategies are optimal.

The remainder of the paper is organized as follows. In Section 2 we describe a general method for establishing upper bounds on the update complexity for geometric problems. This approach is then used in Sections 3 and 4 where update optimal strategies for finding maximal points and points on the convex hull are given, respectively.

2. Preliminaries

We begin by giving some definitions. An input instance is specified by a set P of points in $\Re^2$, and associated with each point $p \in P$, is an area A_p which includes p. Let $S \subseteq P$ be a set of points with some property φ, such as the set of all maximal points in P, or the set of points on the convex hull of P. For convenience, we say that $p \in P$ has the property φ if $p \in S$, and that A_p has the property φ if p has. The algorithm is given only the set $\{A_p | p \in P\}$, and must return all areas which have the property φ.

In order to determine the areas with property φ, the algorithm may *update* an area A_p and determine the exact location of the point p with which it is associated. This reduces A_p to a *trivial* area containing only one point, namely p. The performance of an algorithm is measured in terms of the number of updates it performs to compute the answer; in particular, it should be expressible as a function of the minimal number of updates *OPT* required to verify the solution. An update strategy is said to be *c-update optimal* if it updates at most $c * OPT + O(1)$ inputs, and *update-optimal* if it is c-update optimal for some constant $c \geq 1$. Note that the algorithm may choose to return areas that are not updated as part of the solution. Indeed, in some instances, an algorithm may not need to update *any* areas to solve the problem.

In Sections 3 and 4 we give two update-optimal algorithms for the maximal points and convex hull problems, respectively. These two algorithms are instances of the same generic algorithm, the *witness algorithm*. In this section we describe this algorithm. We begin by giving some definitions. For a given set of areas $\mathcal{F} = \{A_1, \ldots, A_n\}$ we call $C = \{p_1, \ldots, p_n\}$ a *configuration* for $\mathcal{F}$ if $p_i \in A_i$ for $i = 1, \ldots, n$.

Definition 1. Let $\mathcal{F}$ be a set of areas, $A \in \mathcal{F}$ and $p \in A$. Then:

- The point p is ***always*** φ if for any configuration C of $\mathcal{F} - \{A\}$, the point p has the property φ in $C \cup \{p\}$.
- The point p is ***never*** φ if for any configuration C of $\mathcal{F} - \{A\}$, the point p does not have the property φ in $C \cup \{p\}$.
- The point p is ***dependent*** φ if for at least one configuration C of $\mathcal{F} - \{A\}$, the point p has the property φ in $C \cup \{p\}$ and p is not always φ.
- A dependent φ point p ***depends*** on an area B, if for at least one configuration C of $\mathcal{F} - \{A, B\}$, there exist points b_1 and b_2 in B such that the point p has the property φ in $C \cup \{p, b_1\}$ and does not have the property φ in $C \cup \{p, b_2\}$.

We extend this notion from points to areas.

Definition 2. Let $\mathcal{F}$ be a set of areas. Let A be an area in $\mathcal{F}$. We say:

— The area A is ***always*** φ if every point in A is always φ.
— The area A is ***partly*** φ if A contains at least one always φ point and A is not always φ.
— The area A is ***dependent*** φ if A contains at least one dependent φ point and A is not partly φ.
— The area A is ***never*** φ if every point in A is never φ.

Definition 3. Let $\mathcal{F}$ be a set of areas and let $\mathcal{C}$ be a set of configurations for $\mathcal{F}$. A set W of areas in $\mathcal{F}$ is called a *witness set* of $\mathcal{F}$ with respect to C if for any configuration in $\mathcal{C}$ at least one area in W must have been updated to verify the solution. We say W is a witness set of $\mathcal{F}$ if W is a witness set of $\mathcal{F}$ with respect to all possible configurations of $\mathcal{F}$.

The witness algorithm is as follows:

step 1: **while** (there exists at least one partly φ or one dependent φ area)
step 2: **if** there exists a partly φ area
step 3: find a witness set W; update all areas in W.
step 5: **else** // there must exists a dependent φ area //
step 6: find a witness set W; update all areas in W
step 8: **end**

The split in these two cases helps to identify witness sets.

Note that the idea is to concentrate first on partly φ areas and witness sets concerning these areas. Only if there are no partly φ areas left in the given instance, will a strategy based on the existence of dependent areas be used to find witness sets.

Lemma 1. *If there exists a constant k such that every witness set that gets updated by the witness algorithm is of size at most k, then the witness algorithm is k-update optimal.*

Proof. Let F_0 be the set of areas. Assume that the witness algorithm updates n witness sets to determine all areas in F_0 that have property φ. Let $W_1, \ldots, W_n$ be the witness sets in same order as updated by the witness algorithm. Let F_j for $j \leq n$ be the set of areas after updating $W_1, \ldots, W_j$. Further let $\mathcal{C}_\rangle$ be the set of all possible configuration of F_i. So $\mathcal{C}_0 \supset \mathcal{C}_1 \supset \cdots \supset \mathcal{C}_n$, and W_i is a witness set of F_j for $j < i$ and in particular all W_i are witness sets of F_0. Since all W_i are disjoint, at least n updates have to be made to verify the solution ($OPT \geq n$). Since all witness sets are of size at most k, the witness algorithm updated at most $k * n$ areas. Hence the witness algorithm is k-update optimal. □

3. Maximal points

We now give an update-optimal algorithm for the maximal points problem when the areas are either trivial or closures of connected, open areas. We then note (Figure 2) that there is no update-optimal algorithm for arbitrary areas.

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ we say that $p > q$ if $p \neq q$ and $p_x \geq q_x$ and $p_y \geq q_y$. We say a line l *intersects* an area A if they share a common point. We say a line l *splits* an area A if $A - l$ is not connected.

Lemma 2. *Let p be a dependent maximal point. Let Y be the set of areas p depends on. Then all areas of Y must be updated to show p is maximal.*

Proof. Let B be an area on which p depends. Hence there exist $b_1, b_2 \in B$ with $b_1 \not> p$ and $b_2 > p$. In order to verify that p is maximal the area B must be updated. □

Lemma 3. *Let A be a partly maximal area, then there exists a witness set of size at most* 2.

Proof. Since the area A is partly maximal it contains at least one always maximal point. It also contains either a never maximal point or a dependent maximal point. We look at these cases separately.

Case a: *The area A contains a never maximal point.* In order to verify that A is maximal or not, we have to update the area A. Therefore the set $\{A\}$ is a witness set.

Case b: *The area A contains a dependent maximal point p.* Since the area A is partly maximal, by updating only areas other than A the area A can change its status to only an always maximal area. Let B be an area on which p depends. By Lemma 2, in order to verify that p is maximal, we have to at least update B. Hence $\{B, A\}$ is a witness set of size 2. □

Lemma 4. *Let l be a horizontal or vertical line. Further let l split two areas A and B. Then the areas A and B cannot both be always maximal areas.*

Proof. Since all areas are closures of connected open areas the intersection of l with A must contain an open interval and similarly with B. So there exists $a \in A \cap l$ and $b \in B \cap l$ with $a \neq b$. Therefore either $a < b$ or $b < a$. Hence not both A and B are always maximal areas. □

Lemma 5. *If there are no partly maximal areas, but there exists a dependent area, then there exists a witness set of size at most* 3.

Proof. Let A be the area with a dependent maximal point $p \in A$, such that there is no dependent maximal point q in any area such that $q > p$. In other words, p is maximal among the all dependent maximal points. Note that p must exist since all areas are closed. Let l_1 be the vertical line starting at p and going upwards. Let l_2 be the horizontal line starting at p and going to the right. Since p is maximal among the dependent points, every point that lies in $l_1 \cup l_2 - \{p\}$ and also in an area other than A is an always maximal point. Let Q be the top right quadrant of l_1 and l_2 including l_1 and l_2 but not $\{p\}$. Since p is dependent there exists at least one area B with a point in Q and a point not in Q. Since p is maximum among the dependent points every point in Q is always maximal. By our assumption that there are no partly maximal areas the area B must be always maximal.

Hence it does not contain p. Let l_1' be the line l_1 shifted by ε to the left and let l_2' be the line l_2 shifted downwards by ε, where $\varepsilon > 0$ is arbitrarily small. Since all areas are the closure of connected, open areas B must be split either l_1' or l_2'. By Lemma 4 there are at most two areas that contain a point in Q as well as a point not in Q. We call these areas B and C and let $W = \{B, C, A\}$. Note, it is possible that only one of the areas B or C exists. To verify whether A is an always maximal area or not we have at least to update A or we have to verify that p is or is not maximal. For this we have to either update B or C or both. In all cases W is a witness set. □

We now show:

Theorem 1. *Under the restriction to the closure of open, connected areas or trivial areas, the witness algorithm for the maximal point problem is* 3*-update optimal. Furthermore, this is the best possible.*

Proof. The above lemmas show that there is always a witness set of size less than 3. By Lemma 1 we have that the witness algorithm is 3-update optimal. We now argue that there is no c-update optimal algorithm for any $c < 3$.

Consider the three areas A, B and C in Figure 1. The areas A and C are always maximal areas, but updates are needed to determine whether the point of area B is maximal. Note that this picture demonstrate a situation with no partly maximal areas, but with one dependent area and therefore is an example of Lemma 5. For any strategy S in updating these areas there exists a configuration of points for these three areas such that S requires three update where actually only one was needed. For example, consider that S updates A first. We choose one of (1) and (3) to be the input. If the algorithm updates B next, we say that (1) is the input, and force the algorithm to update C as well. However, both (1) and (3) can be "solved" by updating C and B, respectively. To show the lower bound, we simply repeat this configuration arbitrarily often, with each configuration lying below and to the right of the previous one. □

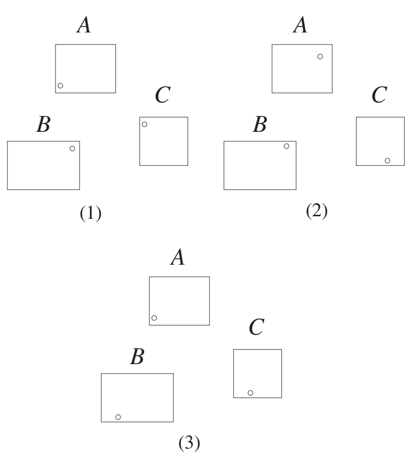

Fig. 1. The three configurations.

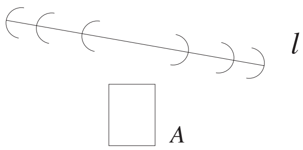

Fig. 2. Example showing that update-optimality is impossible for unrestricted areas.

Remark 1. Finally, we note that there is no update-optimal algorithm for the maximal problem with arbitrary areas. Consider the situation shown in Figure 2. Assume there are n intervals on the line segment l such that after a vertical projection each interval contains the projection of A. Hence each interval is maximal and for any strategy that determines whether A is maximal we have to update some of these intervals. For any given strategy there exists a configuration such that the first $n - 1$ updated intervals correspond to points whose projection on the x-axis lies to the left of the projection of A on the x-axis, and the nth updated interval corresponds to a point such that no point in A can be maximal. If this interval would have been updated first the status of A could have been determined by only one update. Since n was arbitrary there exists no c-update optimal algorithm for arbitrary areas.

4. Convex Hull

Based on the witness algorithm, described in Section 2, we now give an algorithm for computing the convex hull. Again, we restrict the areas to either trivial areas or closures of connected open areas. We also require that every non-empty intersection of two non-trivial areas contains an ε-ball. This algorithm will again be 3-update optimal, and we will also show that this is the best possible. Let $\mathcal{F}$ be the set of areas.

Definition 4. Let $\mathcal{F}$ be a set of areas in $\Re^2$ and let l be a line. Then l splits $\Re^2$ in three regions: two half-planes (H_1, H_2) and l. We say l has an *empty half* with respect to $\mathcal{F}$ if H_1 or H_2 does not intersect any area in $\mathcal{F}$.

Proposition 1. *Let p be a point in $A \in \mathcal{F}$. The point p is always on the convex hull if, and only if, for every configuration with p as the point of the area A, there exists a line l through p such that l has an empty half with respect to $\mathcal{F} - \{A\}$.*

Proof. Obvious. □

Lemma 6. *Let A and B be two non-trivial areas in $\mathcal{F}$ with a non-empty intersection. Further, let $\mathcal{F} - \{A, B\}$ contain at least two areas C and D such that there exists $c \in C$ and $d \in D$ with $c \neq d$. Then neither A nor B is an always convex hull area.*

Proof. By our general condition on all areas in $\mathcal{F}$ the intersection of A and B must contain a non-empty open area E. Since $c \neq d$ and E is open there exist $p, q \in E$ such

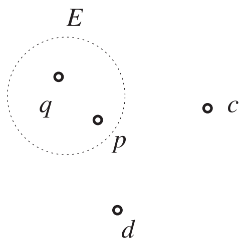

Fig. 3. Choice of p and q.

that p lies inside the triangle with vertices q, c and d, see Figure 3. Hence neither A nor B is an always convex hull area. □

Lemma 7. *Let p be a point in an area $A \in \mathcal{F}$ and let l be a line through p such that in one direction of p the line l splits an area $B \in \mathcal{F} - \{A\}$ and in the other direction of p the line l intersects an area $C \in \mathcal{F} - \{A, B\}$. Further, let $D \in \mathcal{F} - \{A, B, C\}$ such that either*

(i) *$D - \{p\}$ is not empty, if the area C is non-trivial,*
(ii) *$D - l$ is not empty, if the area C is trivial.*

Then p is not always on the convex hull.

Proof. (i) By our assumption there exists a point $d \in D$ with $d \neq p$. Since C is non-trivial it must be the closure of an open area. Hence there exists a $c \in C$ such that p, c and d are not in a line and there exists a point $b \in B$ such that p lies inside the triangle with vertices b, c and d.

(ii) The area C is trivial. So let c be the point in C. Let $d \in D - l$. Since l splits B there exists a point $b \in B$ such that p lies insid the triangle with vertices b, c and d, see Figure 4. □

Lemma 8. *Let p be a dependent convex hull point in an area $A \in \mathcal{F}$. Then there exists a line l through p such that in one direction of p the line l splits an area $B \in \mathcal{F} - \{A\}$ and in the other direction of p the line l intersects an area $C \in \mathcal{F} - \{A, B\}$ and there exists an area $D \in \mathcal{F} - \{A, B, C\}$ such that*

(i) *$D - \{p\}$ is not empty, if the area C is non-trivial,*
(ii) *$D - l$ is not empty, if the area C is trivial.*

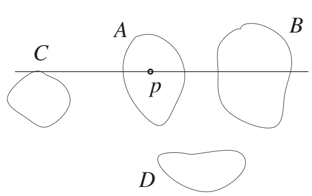

Fig. 4. A line through p that splits an area on one side and touches another one on the other side.

Proof. Since p is a dependent convex hull there exists a configuration G and a line l such that $p \in l$ and l has an empty half H with respect to G. If all areas except A do not intersect with H then l has an empty half for any configuration and p would be therefore an always convex hull. Since all areas are connected, l splits at least one area other than A. By rotating l at the point p we can assume that l in one direction of p splits an area $B \in \mathcal{F} - \{A\}$ and intersects in the other direction another area $C \in \mathcal{F} - \{A, B\}$.

(i) Since p is a dependent convex hull there exist three points in three different areas other than A, such that p lies inside the triangle created by these three points. Hence there exist at least three areas which are not identical to $\{p\}$. Hence if C is not trivial there exists $D \in \mathcal{F} - \{A, B, C\}$ such that $D - \{p\}$ is not empty.

(ii) If C is trivial and all areas except A, B are lying completely on l then in any configuration G_1 with p as the point of the area A, all points in G_1 lie on the convex hull, since all points in G_1 except the point of the area B lie on one line. This contradicts the assumption that p is a dependent convex hull. Hence if C is trivial there exists at least one other area in $\mathcal{F} - \{A, B\}$ which does not lie completely on l. □

Lemma 9. *If there exists a partly convex hull area, then there exists a witness set of size at most* 3.

Proof. Let A be a partly convex hull area. If we assume there exists a never convex hull point in A, then in order to verify whether A is on the convex hull we have to update A. So $\{A\}$ is a witness set. For the rest of this proof let A contain a dependent convex hull point p.

In a verification that shows whether A is on the convex hull and does not update A, the status of the point p must change through updating other areas than A to an always convex hull point.

By Lemma 8 there exists a line l through p such that in one direction of p the line l splits an area $B \in \mathcal{F} - \{A\}$ and in the other direction of p the line l intersects an area $C \in \mathcal{F} - \{A, B\}$ and there exists an area $D \in \mathcal{F} - \{A, B, C\}$ such that

(i) $D - \{p\}$ is not empty, if the area C is non-trivial,
(ii) $D - l$ is not empty, if the area C is trivial.

We look at these cases separately.

(i) C is non-trivial. Since A is a partly convex hull and connected, there exists a point $q \in A$ and a line l' through q such that in one direction of q the line l splits the area B and in the other direction of q the line l intersects C. By Lemma 7 without updating B or C, the points p and q cannot both change their status to always convex hull points. Hence $\{A, B, C\}$ is a witness set.

(ii) C is trivial. By Lemma 7 without updating B or D the point p cannot change its status to an always convex hull point. Hence $\{A, B, D\}$ is a witness set. □

The following definitions and lemmas lead to Theorem 2, which shows that we can also find a witness of size at most 3 if there are no partly convex hull areas but at least one dependent area.

Definition 5. Let $\mathcal{F}$ consist only of always convex hull areas. Let A and B be two areas in $\mathcal{F}$. We call A and B neighbors if in any configuration the points of A and B are adjacent on the convex hull.

Lemma 10. *Let $\mathcal{F}$ consist of only always convex hull areas and let $\mathcal{F}$ not contain identical trivial areas. Further, let $\mathcal{F}$ contain three or more areas. Then every area in $\mathcal{F}$ has exactly two neighbors.*

Proof. Since $\mathcal{F}$ has no multiple trivial areas and every non-trivial area is a closure of an open area there exists a configuration C_1 of $\mathcal{F}$ such that all points in C_1 are distinct. Let A be an area in $\mathcal{F}$ and let a be the point in C_1 that corresponds to the area A. Since all points in C_1 are distinct the point a has exactly two neighbors ($b \in B \in \mathcal{F}$ and $c \in C \in \mathcal{F}$) on the convex hull with respect to C_1. We will show that the areas A and B as well as A and C are neighboring areas.

Let us assume that A and B are not neighboring areas. So, there exists an area $D \in \mathcal{F}$ and a configuration C_2 with $a' \in A$, $b' \in B - \{a'\}$ and $d' \in D - \{a', b'\}$ such that a' and d' are neighbors and a' and b' are not. Since all of these points are disjoint we can change our set of areas as follows.

$$\mathcal{F}' = \left\{ S - \bigcup_{T \in \{A,B,C,D\}, T \text{ is trivial}, S \neq T} T\colon \ S \in \{A, B, C, D\} \right\}.$$

Since all areas in $\mathcal{F}$ are either closures of connected open areas or trivial, all areas in $\mathcal{F}'$ are connected. Note also that by Lemma 6 none of the areas in $\mathcal{F}'$ intersects with another area in $\mathcal{F}'$. Let d be the corresponding point of the area D in the configuration C_1 and let c' be the corresponding point of the area C in the configuration C_2. Note that C_1 and C_2 when restricted to points corresponding to the areas A, B, C and D are configuration for $\mathcal{F}'$. Since the areas A, B, C and D are connected we can continuously transform the configuration C_1 (see Figure 5) to the configuration C_2 (see Figure 6). Since all areas in $\mathcal{F}'$ are disjoint and always convex hulls, this is not possible; a contradiction. Hence A and B are neighboring areas. Similarly for the areas A and C. □

Definition 6. Let $\mathcal{F}$ be a collection of areas. Let A and B be in $\mathcal{F}$. Let $\mathcal{F}' = \{C \in \mathcal{F} : C$ is always convex hull$\}$. We call A and B *always neighbors* if A and B are always convex hull areas and A and B are neighbors in $\mathcal{F}'$.

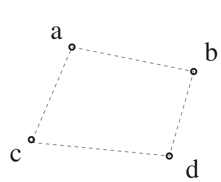

Fig. 5. Configuration where a and b are neighbors.

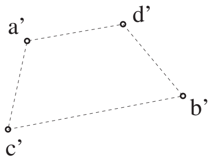

Fig. 6. Configuration where a' and b' are neighbors.

Definition 7. Let A and B be always neighbors. We call the set

$$N_{A,B} = \{p : \exists q \in A, r \in B \text{ with } p \in \text{edge}_{q,r}\}$$

the *neighboring band* of A and B.

Lemma 11. *If there are no partly convex hull areas in $\mathcal{F}$, but $\mathcal{F}$ contains at least three always convex hull areas, then any dependent convex hull area $A \in \mathcal{F}$ must intersect with at least one neighboring band.*

Proof. Since we have at least three always convex hull areas, the neighboring bands of these areas build a closed "polygon" P. Note that an edge of P is a neighboring band and it might therefore be thick. We call all points that lie either outside of P or on the convex hull of P *outer points*, points that lie in a neighboring band except on the convex hull of P, points on P and the rest *inner points*. Since all areas in $\mathcal{F}$ are closed, any area that contains outer points must contain an always convex hull point, which contradicts our assumption that there are no partly convex hull areas. In particular, no dependent convex hull area contains outer points. Since all inner points are never convex hull points, a dependent convex hull area must contain at least a point on P and therefore it must intersect with at least one neighboring band. □

Lemma 12. *Let there be no partly convex hull areas in $\mathcal{F}$. Let $\mathcal{F}$ contain at least three always convex hull areas. If A is a dependent convex hull area intersecting the neighboring band $N_{B,C}$, then $\{A, B, C\}$ is a witness set.*

Proof. Assume that we do not need to update A, B or C. Since A is a dependent convex hull area, in the worst case we must update other areas to change the status of A. Since $\mathcal{F}$ contains at least three always convex hull areas, without updating B or C the area A will not become an always convex hull area. Hence we must update an area D such that A becomes a never convex hull area. Therefore the updated trivial area D' must intersect $N_{B,C}$ and it must be a dependent convex hull area. We are now in a similar situation as before. Since we do not update B or C we have to update an area E such that D' becomes a never convex hull area, but again the updated trivial area E' lies in $N_{B,C}$. Therefore eventually we have to update the area B or C. □

Lemma 13. *If there are no partly convex hull areas in $\mathcal{F}$ and $\mathcal{F}$ contains at least one dependent convex hull area, there exists a witness set of size at most* 3.

Proof. *Case* 1: *$\mathcal{F}$ contains one always convex hull area A*. If the area A is not updated other updates must lead to the situation where there are only three areas in the collection of areas without counting multiple trivial area. Hence A and any two non-trivial areas in $\mathcal{F}$ form a witness set.

Case 2: *$\mathcal{F}$ contains two always convex hull areas A and B*. Similar to Case 1. If the areas A and B are not updated, other updates must lead to the situation in which there are only three areas, without counting multiple trivial areas. Hence A, B and any non-trivial area in $\mathcal{F}$ form a witness set.

Case 3: *$\mathcal{F}$ contains three or more always convex hull areas*. Let D be a dependent convex hull area in $\mathcal{F}$. By Lemma 11 the area D must intersect at least the neighboring band $N_{B,C}$ where B and C are always convex hull areas in $\mathcal{F}$. By Lemma 12 the set $\{D, B, C\}$ is a witness set. □

Theorem 2. *Under the restriction to the closure of open*, *connected areas or trivial areas*, *and every non-empty intersection of two non-trivial areas contains an ε-ball*, *the witness algorithm for the convex hull problem is* 3*-update optimal*. *Furthermore*, *this is the best possible*.

Proof. By Lemmas 13, 9 and 1 the witness algorithm is 3-update optimal.

Figure 7 shows that there is no algorithm which is c-update optimal for $c < 3$. The areas A, B, D and E are always on the convex hull. However, for any strategy to determine the status of C there exists a configuration such that this strategy requires to update all three areas, but starting with updating the last one would have given the answer directly. Since we can construct a set of areas that consists entirely of triples following the same pattern, there cannot exist a c-update optimal algorithm with c less than 3. □

Remark 2. Using a similar ideas to that described in Remark 1 we can show that there is no update-optimal algorithm for the convex hull problem with arbitrary areas:

In Figure 8 let there be n intervals on the line l, let each interval contain at least two points q and q' such that p lies inside the triangle r, q, s and p lies outside the triangle r, q', s. In order to determine whether p is on the convex hull the intervals on the line l must be updated. For any strategy there exists a configuration such that the first $n - 1$ updates of intervals on l do not determine whether p is on the convex hull or not and the nth interval updated ensures that p cannot lie on the convex hull. Hence there exists no update-optimal algorithm for arbitrary areas.

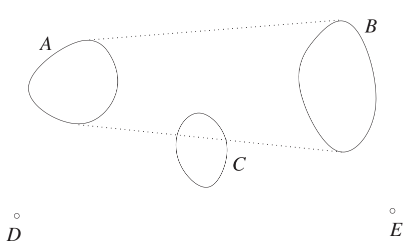

Fig. 7

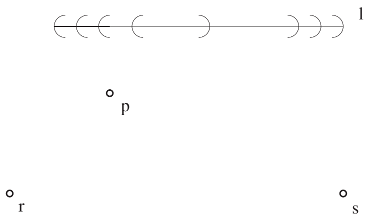

Fig. 8

References

[1] J. Basch, L. Guibas and J. Hershberger, Data Structures for Mobile Data, *J. Algorithms* 31:1–28, 1999.

[2] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, 1998.

[3] T. Feder, R. Motwani, R. Panigrahy, C. Olston and J. Widom, Computing the Median with Uncertainty, *Proc*. 32*nd Symposium on Theory of Computing*, pp. 602–607, ACM Press, New York, 2000.

[4] T. Feder, R. Motwani, L. O'Callaghan, C. Olston and R. Panigrahy, Computing Shortest Paths with Uncertainty, *Proc*. 20*th International Symposium on Theoretical Aspects of Computer Science*, pp. 355–366, LNCS 2607, Springer, Berlin, 2003.

[5] S. Kahan, A Model for Data in Motion, *Proc*. 23*rd Symposium on Theory of Computing*, pp. 267–277, ACM Press, New York, 1991.

[6] S. Khanna and W.-C. Tan, On Computing Functions with Uncertainty, *Proc*. 20*th Symposium on Principles of Database Systems*, pp. 171–182, ACM Press, New York, 2001.

[7] C. Olston and J. Widom, Offering a Precision-Performance Tradeoff for Aggregation Queries over Replicated Data, *Proc*. 26*th International Conference on Very Large Data Bases*, pp. 144–155, Morgan Kaufmann, San Mateo, CA, 2000.

Received in final form January 6, 2004. *Online publication January* 28, 2005.