

FAST SEARCHING IN A REAL ALGEBRAIC MANIFOLD
WITH APPLICATIONS TO GEOMETRIC COMPLEXITY

by

Bernard Chazelle

Abstract:

This paper generalizes the multidimensional searching scheme of Dobkin and Lipton [SIAM J. Comput. 5(2), pp. 181-186, 1976] for the case of arbitrary (as opposed to linear) real algebraic varieties. Let d, r be two positive constants and let P_1, \dots, P_n be n rational r -variate polynomials of degree $\leq d$. Our main result is an $O(n^{2r+6})$ data structure for computing the predicate $[\exists i (1 \leq i \leq n) \mid P_i(x) = 0]$ in $O(\log n)$ time, for any $x \in E^r$. The method is intimately based on a decomposition technique due to Collins [Proc. 2nd GI Conf. on Automata Theory and Formal Languages, pp. 134-183, 1975]. The algorithm can be used to solve problems in computational geometry via a *locus* approach. We illustrate this point by deriving an $o(n^2)$ algorithm for computing the time at which the convex hull of n (algebraically) moving points in E^2 reaches a steady state.

This research was supported in part by NSF grants MCS 83-03925 and the Office of Naval Research and the Defense Advanced Research Projects Agency under contract N00014-83-K-0146 and ARPA Order No. 4786.

1. Introduction

Let F be a family of n hyperplanes in E^r , with r taken as a constant. Dobkin and Lipton [DL] have shown how to represent F , using a polynomial amount of storage, so that whether a given point lies in any of the hyperplanes of F can be checked in $O(\log n)$ time. Our goal is to generalize the technique used to achieve this result for the case where F is a family of algebraic varieties. Put more formally, let d, r be two positive constants and let $F = \{P_1, \dots, P_n\}$ be a set of n polynomials of degree $\leq d$ in r real variables with rational coefficients. We consider the problem of preprocessing F so that, for any $x \in E^r$, the predicate $[\exists i (1 \leq i \leq n) \mid P_i(x) = 0]$ can be evaluated efficiently. If the predicate is true, any one of the indices i for which $P_i(x) = 0$ should be reported — note that requiring the report of all such indices might by itself rule out a fast response. If the predicate is false, the point x lies in an open, maximally connected region, over which the value of each P_i keeps a constant sign. Assuming that these regions have been labelled in preprocessing, retrieving the label corresponding to the region containing x will also be required. In this form, the problem is a direct generalization of the well-known *planar point location* problem. Previous work on point location with non-linear boundaries has been limited to the case $r = 2$, culminating in the optimal algorithm of Edelsbrunner, Guibas and Stolfi [E].

Our main result is a data structure for answering any such query in $O(\log n)$ time. The space and time necessary to construct the data structure are in $O(n^{2^{r+6}})$, i.e. polynomial in n . The main consequence of our result is to open the door to *locus*-based methods for solving previously untouchable problems of computational geometry: the locus approach for retrieval problems involves considering each query as a higher-dimensional point and partitioning the underlying space into regions providing the same answer — see [O] for a discussion of this approach. We illustrate this notion on a specific example by considering a problem posed by Atallah in [A]. Given n points in E^2 , each moving as a fixed degree polynomial function of time, is it possible to compute in $o(n^2)$ time the first instant at which the convex hull of the points will enter its final (steady) configuration? We use our generalized point location algorithm to solve this problem in the affirmative.

In the next section (Section 2), we review the necessary algebraic tools and describe the point

location algorithm in Section 3. As in Dobkin and Lipton's method, the search proceeds by iterated projections on canonical hyperplanes. The preprocessing is inspired by Collins' cylindrical algebraic decomposition [C], adding some refinements due to Schwartz and Sharir [SS]. Ultimately, the spirit of the method goes back to Tarski's fundamental work on the decidability of elementary algebra [T]. In the remainder of the paper (Sections 4–5), we present our solution to Atallah's problem and conclude with directions for further research.

2. The Algebraic Machinery

Most of the algebraic notions involved in this work can be found exposed in great detail in two milestone papers by Collins [C] and Schwartz and Sharir [SS]. We have tried to adhere to the terminology used in these papers as much as possible. The fundamental algebraic concepts can be found in van der Waerden's classic text [W]; for the specialized treatment of resultants and subresultants used in the paper the reader should turn to Brown and Traub [BT].

I) Collins' Decidability Theorem

In 1948, Tarski proved that every statement in elementary algebra (i.e. in the elementary theory of real-closed fields) is decidable. The non-elementary procedure given by Tarski was subsequently (computationally) improved in a number of different ways by several researchers (e.g. Seidenberg, Cohen, Collins, Monk/Solovay, Ben-Or/Kozen/Reif). For the purpose of the present work, we shall use Collins' decision procedure as a guiding framework. Let a *standard prenex formula* be any logical sentence of the form $(Q_k x_k)(Q_{k+1} x_{k+1}) \dots (Q_r x_r) \phi(x_1, \dots, x_r)$, where Q_i is the universal or existential quantifier and $\phi(x_1, \dots, x_r)$ is a quantifier-free formula made of boolean connectives, standard comparators, and polynomials with rational coefficients in the real variables x_1, \dots, x_r . A logical sentence is called an *atomic formula* if it is free of quantifiers and logical connectives.

Theorem 1. (Collins [C]) Let Φ be an arbitrary standard prenex formula with r variables, c atomic formulas, m polynomials of degree $\leq d$, and no integral coefficient of length $\geq n$. Whether Φ is true or false can be decided in $cn^3(2d)^{2^{r+c}} m^{2^{r+c}}$ operations.

II) The Cylindrical Algebraic Decomposition

A *Collins decomposition* of E^r is a scheme for partitioning E^r in order to discriminate among the connected regions of E^r induced by a real algebraic variety. Before reviewing the main components of Collins' technique, we need define the fundamental notion of *cylindrical algebraic decomposition* (c.a.d., for short). A c.a.d. of E^r is a partitioning of E^r defined recursively as follows. For $r = 1$, a c.a.d. is a finite set of disjoint open intervals, along with the algebraic numbers bounding them, whose union form E^1 . For $r > 1$, a c.a.d. K is defined in terms of a c.a.d. K' of E^{r-1} and an r -variate polynomial $P(x_1, \dots, x_{r-1}, y)$ with rational coefficients. Let $K' = \{c_1, \dots, c_\mu\}$; for each $c_i \in K'$, there exists ν_i such that for each $x = (x_1, \dots, x_{r-1}) \in c_i$, $P(x, y)$, regarded as a polynomial in y , has ν_i real roots $f_{i,1}(x) < \dots < f_{i,\nu_i}(x)$, each of which is a continuous function in x over c_i . If $\nu_i = 0$, put $c_{i,1} = c_i \times E^1$. If $\nu_i > 0$, set $c_{i,2j} = \{(x, f_{i,j}(x)) \mid x \in c_i\}$ for $1 \leq j \leq \nu_i$, set $c_{i,2j+1} = \{(x, y) \mid x \in c_i \ \& \ f_{i,j}(x) < y < f_{i,j+1}(x)\}$ for $1 \leq j < \nu_i$. Also, put $c_{i,1} = \{(x, y) \mid x \in c_i \ \& \ y < f_{i,1}(x)\}$ and $c_{i,2\nu_i+1} = \{(x, y) \mid x \in c_i \ \& \ f_{i,\nu_i}(x) < y\}$. Finally, K is defined as the set of cells $\{c_{1,1}, \dots, c_{1,2\nu_1+1}, \dots, c_{\mu,1}, \dots, c_{\mu,2\nu_\mu+1}\}$. Informally, the cells can be formed by considering the cylinders based at each $c \in K'$ and chopping them off with the real hypersurface $P(x_1, \dots, x_{r-1}, y) = 0$. P (resp. K') is called the base polynomial (resp. cylindrical algebraic decomposition) of K . Since a c.a.d. is defined in terms of a unique base c.a.d. of lesser dimension, by induction, K defines an *induced* c.a.d. for each E^k ($1 \leq k < r$). Incidentally, one should note that each cell of K is "well-behaved," in the sense that it is topologically equivalent to an open cell of dimension $\leq r$.

To be of interest here, a cylindrical algebraic decomposition must provide a framework for discriminating among several algebraic varieties. So, let F be a family of n functions of r variables. We say that a c.a.d. K is *F-invariant* if for each $c \in K$ and each $f \in F$, we have: $f(x) = 0$ for each $x \in c$, $f(x) < 0$ for each $x \in c$, or $f(x) > 0$ for each $x \in c$. To prevent topological anomalies and thus facilitate the computation, we may enforce a cylindrical algebraic decomposition to be *well-based*. K is said to be well-based if its base polynomial $P(x, y)$, regarded as a polynomial in y , is not identically zero for any given value of x in E^{r-1} . With these conditions, Schwartz and Sharir [SS] have shown that each root function $f_{i,j}$ (defined over $c_i \in K'$) can be extended continuously

over the closure of c_i . This implies that the closure of every cell in K is a union of cells. Informally, this fairly intuitive result prevents K from displaying any pathological configuration. In particular, this means that every line $(x_1, \dots, x_{r-1}) \times E^1$ intersects the algebraic variety $P(x, y) = 0$ a finite number of times. These intersections will form the basis of the binary search underlying the point location algorithm to be presented in the next section.

Following Collins' terminology, an *algebraic sample* of K is a set of points with algebraic coordinates, one in each cell of K (recall that a number is algebraic if it is a root of a polynomial). An algebraic sample is *cylindrical* (c.a.s) if 1) $r = 1$, or 2) the set of $r - 1$ first coordinates of each point form a c.a.s of K' , the base c.a.d. of K . If $\{c_{i,1}, \dots, c_{i,2\nu_i+1}\}$ is the set of cells of K associated with the cell c_i of K' , the sample points in each $c_{i,j}$ ($1 \leq j \leq 2\nu_i + 1$) all share the same $r - 1$ first coordinates. We conclude this string of definitions with a word on the representation of a c.a.d. The *standard definition* of a c.a.d. $K = \{d_1, \dots, d_\nu\}$ is a sequence of quantifier-free formulas $\{\phi_1(x), \dots, \phi_\nu(x)\}$ ($x = (x_1, \dots, x_r)$), where $\phi_i(x)$ has only x_1, \dots, x_r as free variables and $d_i = \{x \in E^r \mid \phi_i(x) \text{ is true}\}$. $\phi_i(x)$ is called the *defining* formula of d_i .

III) The Collins Construction

Let d, r be two positive constants and let $F = \{P_1, \dots, P_n\}$ be a set of n polynomials of degree $\leq d$ in r real variables with rational coefficients. A Collins decomposition, K , is an F -invariant cylindrical algebraic decomposition. The construction of K proceeds recursively. Sharir and Schwartz solve the problem by considering the product $P = \prod_{1 \leq i \leq n} P_i$ and using the well-known fact that P (as a polynomial in x_r) has exactly $d_1 - d_2$ distinct real roots, where d_1 is the degree of P and d_2 is the degree of R , the gcd of P and its x_r -derivative. It then suffices to find a c.a.d. of E^{r-1} such that on each cell the degrees of both P and R remain constant. We follow the same strategy; however, because of its prohibitively high degree, we must avoid considering the product P in the actual construction of the cylindrical algebraic decomposition. Instead, although we will keep P as the base polynomial of the c.a.d., we will use the fact that P is a product of smaller-degree polynomials when carrying out the construction. Intuitively, it appears that taking all pair-products $P_i \times P_j$ should be sufficient, for the number of distinct roots change either when two roots of P_i

“merge” into one or when a root of P_i “merges” with a root of P_j . In the present case, degeneracies among the coefficients (which are, we should recall, polynomials in $r - 1$ variables) necessitates a more careful treatment.

To begin with, we recall basic facts from elimination theory. Let Q be an r -variate polynomial with rational coefficients. We regard $Q(x_1, \dots, x_r) = \sum_{0 \leq i \leq p} Q_i(x_1, \dots, x_{r-1})x_r^i$ as a polynomial in x_r with coefficients in the ring of polynomials in $r - 1$ variables. Let $\deg(Q) = p$ be the degree of Q and $\text{ldcf}(Q) = Q_p(x_1, \dots, x_{r-1})$ be the leading coefficient of Q . Following Tarski and Collins, we define the *reductum* of Q , $\text{red}(Q) = \sum_{0 \leq i \leq p-1} Q_i(x_1, \dots, x_{r-1})x_r^i$. We also introduce $\text{red}^0(Q) = Q$ and for each $k \geq 0$, $\text{red}^{k+1}(Q) = \text{red}(\text{red}^k(Q))$. Finally, let $\text{der}(Q)$ denote the x_r -derivative of Q . Let $A(x)$ and $B(x)$ be two polynomials in the real variable x with $\deg(A) = a$, $\deg(B) = b$. Let $U_j(x)$ and $V_j(x)$ be two polynomials with degrees $b - j - 1$ and $a - j - 1$, respectively. Following [SS], consider the equation $A(x)U_j(x) - B(x)V_j(x) = C_j(x)$ where $C_j(x)$ is a polynomial of degree at most $j - 1$. This system has a non-zero solution if and only if $\text{psc}^j(A, B) = 0$, where $\text{psc}^j(A, B)$ (the j^{th} principal subresultant coefficient of A and B) is the determinant of the $(a + b - 2j) \times (a + b - 2j)$ matrix of the homogeneous system of linear equations obtained by requiring that the polynomial $A(x)U_j(x) - B(x)V_j(x) - \text{red}^{(a+b-2j)}(A(x)U_j(x) - B(x)V_j(x))$ should be identically zero (see [BT] for details). The unique factorization theorem for polynomials implies that A and B have exactly j common roots (i.e. j is the degree of $\text{gcd}(A, B)$) if and only if j is the least index for which $\text{psc}^j(A, B) \neq 0$. This result is at the basis of the recursive construction of an F -invariant c.a.d.

Next, we define what arguments should be passed to the decomposition algorithm at the second recursive call. To do so, we define the *projection* of F , denoted G , as the union of G_2, G_3, G_4 , with $G_1 = \{\text{red}^k(P) \mid P \in F \ \& \ k \geq 0 \ \& \ \deg(\text{red}^k(P)) \geq 1\}$, $G_2 = \{\text{ldcf}(P) \mid P \in G_1\}$, $G_3 = \{\text{psc}^k(P, \text{der}(P)) \mid P \in G_1 \ \& \ 0 \leq k < \deg(\text{der}(P))\}$, $G_4 = \{\text{psc}^k(P, Q) \mid P, Q \in G_1 \ \& \ 0 \leq k < \min(\deg(P), \deg(Q))\}$. The notion of projection generalizes Dobkin and Lipton’s idea of pairing up hyperplanes: the pairing takes place in G_4 , while G_2 and G_3 account for the occasional losses of rank in each variety. The following result is proven in [C]: let K' be a G -invariant c.a.d. of E^{r-1} and let c_i be any cell of K' ; the total number of distinct roots of $P_1(x_1, \dots, x_r), \dots, P_n(x_1, \dots, x_r)$, as

polynomials in x_r , remains constant as (x_1, \dots, x_{r-1}) varies in c_i . These roots form a well-ordered set of continuous functions over c_i : $f_{i,1}(x_1, \dots, x_{r-1}), \dots, f_{i,\nu_i}(x_1, \dots, x_{r-1})$. As a result, for each $c_i \in K'$, the partition of $c_i \times E^1$ induced by the hypersurfaces $f_{i,1}(x) = 0, \dots, f_{i,\nu_i}(x) = 0$ ($x \in E^{r-1}$) defines an F -invariant c.a.d. of E^r .

This provides a recursive scheme for computing an F -invariant c.a.d. K of E^r . The algorithm takes F as input and recurses by calling itself with G , the projection of F , as argument. The output of the algorithm will be a c.a.s. of F , $\{\beta_1, \dots, \beta_\nu\}$, where for each $i = 1, \dots, \nu$ each coordinate of $\beta_i \in E^r$ is represented by a quantifier-free defining formula. It turns out that Collins' sophisticated method for computing a standard definition of K is not really necessary (although doable within the same asymptotic running time). Instead, we need an additional piece of information: a correspondence between sample points and their defining polynomials in F . If $r > 1$, K has a base c.a.d. $K' = \{c_1, \dots, c_\mu\}$ (which is G -invariant). Let $\{\beta'_1, \dots, \beta'_\mu\}$ be the c.a.s. of K' , computed recursively. For each $i = 1, \dots, \mu$, let $\{\beta_{i,1}, \dots, \beta_{i,2\nu_i+1}\}$ be the points of the c.a.s. of K , ordered in ascending x_r -order, whose first $r-1$ coordinates form the point β'_i . Each point $\beta_{i,2j}$ ($1 \leq j \leq \nu_i$) lies on at least one algebraic variety of the form $P_l(x) = 0$. Let $l_{i,j}$ be any such value of l and let $\beta_{i,2j} = (a_1, \dots, a_r)$; we define $m_{i,j}$ as the number of distinct roots of $Q(y)$ that are strictly smaller than a_r , where $Q(y) = P_{l_{i,j}}(a_1, \dots, a_{r-1}, y)$ is regarded as a polynomial in y . As part of the output, we require the sequence $\{(l_{i,1}, m_{i,1}), \dots, (l_{i,\nu_i}, m_{i,\nu_i})\}$ for each $i = 1, \dots, \mu$. This sequence will be necessary later on in order to carry out the binary searches underlying the point location algorithm.

The next step is to show how derive these sequences from $\{\beta_{i,1}, \dots, \beta_{i,2\nu_i+1}\}$ ($1 \leq i \leq \mu$). Recall that the latter sequences are provided directly by the Collins construction. Let $\phi(x)$ be the quantifier-free defining formula for $\beta_{i,2j}$ ($1 \leq j \leq \nu_i$). Trivially, we can test the predicate $[\exists x \in E^r \mid \phi(x) \ \& \ (P_l(x) = 0)]$ for each $l = 1, \dots, n$, and pick as $l_{i,j}$, say, the first value of l found to satisfy the predicate. To obtain $m_{i,j}$, it suffices to express with a prenex formula the fact, F_k , that z is a root of Q and $Q(y) = P_{l_{i,j}}(a_1, \dots, a_{r-1}, y)$ has exactly k distinct roots strictly smaller than z . We can express F_k with the formula $R_{a_1, \dots, a_{r-1}, k}^{l_{i,j}}(z) = [(\exists y_1, \dots, y_k)(\forall x) \mid ((Q(z))^2 + (Q(y_1))^2 + \dots + (Q(y_k))^2 = 0) \ \& \ (y_1 < \dots < y_k < z) \ \& \ (Q(x) \neq 0 \text{ or } z \leq x \text{ or } (y_1 - x)^2 + \dots + (y_k - x)^2 =$

0)]. The value of $m_{i,j}$ is then given by the unique index k for which $R_{a_1, \dots, a_{r-1}, k}^{l_{i,j}}(z)$ is true, with $\beta_{i,2j} = (\beta'_i, z)$. In analyzing the complexity of the algorithm, we assume that only rational symbolic calculations are used during the course of the computation. We omit the proofs of the following complexity results, which can be found in [C]. Let d be the maximum degree of any polynomial in F in any variable and let l be an upper bound on the *norm-length* of any polynomial of F . The norm-length of a polynomial is the number of bits needed to represent the sum of the absolute values of its coefficients. As usual, we assume that r , d , and l are independent of n . The F -invariant c.a.d. produced by the Collins construction consists of $O((2d)^{3^{r+1}} n^{2^r}) = O(n^{2^r})$ cells. The total number of polynomials defined in the various projections introduced in the decomposition is bounded above by $O((2d)^{3^r} n^{2^{r-1}}) = O(n^{2^{r-1}})$ and the degree of each polynomial is at most $\frac{1}{2}(2d)^{2^{r-1}} = O(1)$. The norm-length of each polynomial is $\leq (2d)^{2^r} l = O(1)$. Consider now the c.a.s. of the decomposition. Each algebraic point is represented by its coordinates. We can represent an algebraic number in two ways, depending on the interpretation we give to it. Either it is a real root α of a polynomial $A(x)$ with algebraic coefficients. We isolate the root by specifying an interval I with rational endpoints, so α is represented by the pair (A, I) . The coefficients of A (if non-rational) are represented recursively. In some cases, an algebraic number β will appear as an element of the algebraic number field $Q(\alpha)$ (recall that $Q(\alpha)$ is the intersection of all the extension fields of Q which contain α , or equivalently, the smallest subfield of \mathfrak{R} which contains Q and α). In this case, we represent β as a rational polynomial $B(\alpha)$. The degree of any polynomial used in the definition of all the c.a.s.'s is dominated by $(2d)^{2^{2r-1}} = O(1)$ and more interestingly, the norm-length of each polynomial is $\leq l(2d)^{2^{2r+3}} n^{2^{r+1}} = O(n^{2^{r+1}})$. Implementing the Collins construction proper requires $O(l^3(2d)^{2^{2r+8}} n^{2^{r+6}}) = O(n^{2^{r+6}})$ operations. Using Theorem 1 and the previous upper bounds, it is easy to see that this running time asymptotically dominates the overhead of computing the sequences of the form $\{(l_{i,1}, m_{i,1}), \dots, (l_{i,\nu_i}, m_{i,\nu_i})\}$. We refer the reader to [SS] to see how to ensure that the c.a.d. K and all its induced c.a.d.'s are well-based.

3. The Generalized Point Location Algorithm

Most of the ingredients entering the composition of the algorithm have already been introduced. The data structure $DS(F)$ is defined recursively as follows: it includes 1) $DS(G)$, where G is the projection of F ; 2) a c.a.s. of K ; 3) a set of ν one-word memory cells C_1^f, \dots, C_ν^f (which we conveniently associate with the cells of K). Let C_1^g, \dots, C_μ^g be the memory cells associated with $DS(G)$ (in one-to-one correspondence with the cells of $K' = \{c_1, \dots, c_\mu\}$). Each cell C_i^g ($1 \leq i \leq \mu$) stores a pointer to the sequence $\{l_{i,1}, \dots, l_{i,\nu_i}\}$ previously defined. Recall that the cell C_i^g is associated with $2\nu_i + 1$ cells of K (each projecting exactly on c_i). Let $W_i = \{C_{i,1}^f, \dots, C_{i,2\nu_i+1}^f\}$ be the corresponding memory cells, in ascending x_r -order. Consider the sequence $S_i = \{l_{i,1}, \dots, l_{i,\nu_i}\}$ as an ordered set of *keys*. The possible outcomes of a binary search in this set is a sequence of $2\nu_i + 1$ keys and open intervals, which we put in one-to-one correspondence with W_i . The data structure is now complete, so we can describe the algorithm.

The input is a family of polynomials F , assumed to be preprocessed as previously described. The *generalized point location* problem to be solved can be stated as follows: "given a query point $q = (q_1, \dots, q_r) \in E^r$, compute the index i such that C_i^f corresponds to the unique cell of K that contains q ." Note in passing the practical importance of having a c.a.s. of K . In most cases, indeed, the point location problem arises when one wishes to compute a function f from E^r to some range (e.g. set of integers or reals). The c.a.d. K partitions E^r into regions on which f is invariant. The availability of a c.a.s. allows us to precompute the unique value of f over each such region, thus reducing the original problem of computing $f(q)$ for arbitrary $q \in E^r$ to a generalized point location problem. If $r = 1$, the algorithm is trivial, so assume that $r > 1$. Recursively, we assume that we have available the index of the cell C_i^g that contains (q_1, \dots, q_{r-1}) . Perform a binary search in S_i with respect to x_r , and report the element of W_i corresponding to the result of the search. Without concern for efficiency, we implement the generic comparison against $l_{i,j}$ as the two-fold question:

1. Does $P_{i,j}(q) = 0$?
2. Is q_r strictly larger or smaller than the $(m_{i,j} + 1)$ st real root of $P_{i,j}(q_1, \dots, q_{r-1}, y)$, regarded here as a polynomial in y ?

The latter question is answered by testing the predicate $[(\forall y) | (y > q_r) \text{ or } \neg(R_{q_1, \dots, q_{r-1}, m_{i,j}}^{i,j}(y))]$. For K to be well-based ensures consistency in the search process, i.e. the intersection of the line $(q_1, \dots, q_{r-1}) \times E^1$ and the variety $P_{i,j}$ always consists of a discrete set of points. To analyze the complexity of the algorithm, we use the fact that each polynomial occurring in any projection has degree $O(1)$, so from Theorem 1 it easily follows that any comparison can be decided in constant time. Since the total number of these polynomials is in $O(n^{2^{r-1}})$, so is the length of any sequence over which a binary search is performed. As a result, each binary search requires $O(2^r \log n)$ time. As mentioned earlier, the preprocessing costs are in $O(n^{2^{r+6}})$, both in time and space. We conclude with the main result of this section.

Theorem 2. Let $F = \{P_1, \dots, P_n\}$ be a family of n fixed-degree r -variate polynomials with rational coefficients and let $S = \{x \in E^r \mid \prod_{1 \leq i \leq n} P_i(x) \neq 0\}$. In $O(n^{2^{r+6}})$ time and space, it is possible to compute a set of algebraic points, one in each connected region of S , as well as set up a data structure for computing the predicate $[\exists i (1 \leq i \leq n) \mid P_i(q) = 0]$, for any $q \in E^r$. In $O(2^r \log n)$ time, the algorithm will return an index i such that $P_i(q) = 0$ if such an index is to be found, otherwise it will return the algebraic point associated with the unique region of S that contains q .

4. Applications to Computational Geometry

In [A], Atallah poses the following problem. Consider n points moving in the plane as a polynomial function of the time. What is the first time their convex hull enters a steady-state, i.e. a combinatorially invariant configuration? We will assume that the real roots of any univariate polynomial of fixed degree with real coefficients independent of n can be computed in constant time with any desired precision — note that this assumption does not in any way follow from Collins' theorem. The naive algorithm consists of computing the steady convex hull in $O(n \log n)$ time [A], and then retrieving the first time each point achieves its steady positioning with respect to each edge on the hull. The maximum of all these times provides the desired value. Can this quadratic algorithm be improved? We will show that it can — at least theoretically. More precisely, we will use the generalized point location algorithm of the previous section to produce an $O(n^{2-\epsilon})$ time

algorithm, for a very small positive constant ϵ . Let $V = \{p_1, \dots, p_n\}$ be a set of $n > 2$ points in the Euclidean plane, subject to algebraic motion. This assumes the existence of $2n$ univariate polynomials $p_1^x, p_1^y, \dots, p_n^x, p_n^y$ of degree d with real coefficients, such that for each i ($1 \leq i \leq n$), $p_i^x(t)$ and $p_i^y(t)$ are respectively the x and y coordinates of p_i at time $t \geq 0$. We assume that d as well as all the polynomials' coefficients are independent of n . Let p_{i_1}, \dots, p_{i_k} be the points on the boundary of the convex hull of V at time t , given in clockwise order with $i_1 < \min(i_2, \dots, i_k)$ (if $p_{i_j}, \dots, p_{i_{j'}}$ coincide, their indices appear in the order $i_j < \dots < i_{j'}$). Let $H(t)$ be the (uniquely defined) sequence $\{i_1, \dots, i_k\}$. It is trivial to show that $H(t)$ converges as t grows to infinity. We define the *threshold* of $H(+\infty)$ as the smallest value of $t \geq 0$ such that $[(\forall t' \geq t) \mid H(t) = H(t')]$.

Let $p_i^x(t) = \sum_{0 \leq j \leq d} a_{i,j} t^j$ and $p_i^y(t) = \sum_{0 \leq j \leq d} b_{i,j} t^j$ for $i = 1, \dots, n$. Wlog, assume that $H(+\infty) = \{1, \dots, m\}$ ($m \leq n$) and that all n points $(a_{i,0}, \dots, a_{i,d}, b_{i,0}, \dots, b_{i,d})$ of \mathfrak{R}^{2d+2} are pairwise distinct. In $O(n \log n)$ time, compute $H(+\infty)$ [A] and check all pairs (p_i, p_{i+1}) ($1 \leq i \leq m$) in order to determine the largest $t_0 \geq 0$ such that, for some i , we have $p_i^x(t_0) = p_{i+1}^x(t_0)$ and $p_i^y(t_0) = p_{i+1}^y(t_0)$; note that we may have $t_0 = -\infty$. Here (as in the following), index arithmetic is taken mod m . Similarly, we ensure the convexity of the polygon $\{p_1, \dots, p_m\}$ by considering the function $f_i(q) = (p_i^y(t) - p_{i+1}^y(t))q_x + (p_{i+1}^x(t) - p_i^x(t))q_y + p_i^x(t)p_{i+1}^y(t) - p_i^y(t)p_{i+1}^x(t)$. The point $q = (q_x, q_y) \in E^2$ lies to the right (resp. on, to the left) of the oriented line $(p_i, \overrightarrow{p_i p_{i+1}})$ iff $f_i(q) < 0$ (resp. $f_i(q) = 0, f_i(q) > 0$). For each p_i ($1 \leq i \leq m$) compute the largest real root of $f_{i+1}(p_i)$ as a polynomial in t ; discard every case where the polynomial is identically zero. Let t_1 be the largest value thus obtained (or $-\infty$ if there is none), and let $t_2 = \max(0, t_0, t_1)$. Trivially, t_2 can be computed in $O(n)$ time, once $H(+\infty)$ is available. All that remains to be done is to compute the first instants at which each p_j ($m < j \leq n$) lies inside $H(+\infty)$ for good. To do so, we allow ourselves some preprocessing. Let $q(t) = (q_x(t), q_y(t))$ be a point in E^2 , with $q_x(t) = \sum_{0 \leq j \leq d} q_j^x t^j$ and $q_y(t) = \sum_{0 \leq j \leq d} q_j^y t^j$. The point $\chi = (q_0^x, \dots, q_d^x, q_0^y, \dots, q_d^y)$ belongs to E^{2d+2} and is independent of n . Let $\text{sign}(A) = -1$ (resp. $= 0, 1$) if $A < 0$ (resp. $A = 0, > 0$). We define $t(\chi) = [\min t \in \mathfrak{R} \mid t \geq t_2 \ \& \ (\forall i; 1 \leq i \leq m) (\forall t' > t) \ \text{sign}(f_i(q(t))) = \text{sign}(f_i(q(t')))]$. We next describe a fast algorithm for computing $t(\chi)$ based on the generalized point location algorithm of the preceding section.

For each i ($1 \leq i \leq m$), let $F = \{\phi_1(\chi, t), \dots, \phi_m(\chi, t)\}$, where $\phi_i(\chi, t)$ denotes the $(2d + 3)$ -variate polynomial of degree $2d + 1$, $f_i(\sum_{0 \leq j \leq d} q_j^x t^j, \sum_{0 \leq j \leq d} q_j^y t^j)$. Let K be the F -invariant c.a.d. of E^{2d+3} provided by the procedure described in Section 2, and let $K' = \{c_1, \dots, c_\mu\}$ be its base c.a.d. (i.e. the induced c.a.d. of E^{2d+2}). Recall that for each $c_i \in K'$ the procedure provides us with a sequence of indices (possibly empty) $S_i = \{l_{i,1}, \dots, l_{i,\nu_i}\}$ with the following meaning: for any given $\chi \in c_i$ the line $\chi \times E^1$ provides an increasing sequence of real roots for the univariate polynomials $\phi_{l_{i,1}}(\chi, t), \dots, \phi_{l_{i,\nu_i}}(\chi, t)$. The interpretation of this sequence is trivial: it gives the indices of the lines passing through $p_i p_{i+1}$ that are intersected by the trajectory of χ in chronological order (from $t = -\infty$ to $t = +\infty$). If the sequence is empty, χ never intersects such a line. Once K' has been preprocessed for point location, computing $t(\chi)$ is straightforward. Locate the cell c_i that contains χ and check whether the sequence S_i is empty. If yes, set $t(\chi) = t_2$. If the sequence is not empty, the trajectory of χ intersects the line passing through $p_{l_{i,\nu_i}} p_{l_{i,\nu_i}+1}$ at some time t and does not intersect any other such line subsequently. We obtain t by computing the largest real root of $\phi_{l_{i,\nu_i}}(\chi, t)$ as a polynomial in t (which must exist). Finally we set $t(\chi) = \max(t_2, t)$. From Theorem 2, we immediately conclude.

Lemma 1. In $O(m^{2^{2d+9}})$ time and space, it is possible to construct a data structure so that the function $t(\chi)$ can be evaluated at any point $\chi \in E^{2d+2}$ in $O(4^d \log m)$ time.

We are now ready to attack Atallah's problem. We use a batching strategy inspired by Yao's work on higher-dimensional MST [Y]. Partition the boundary of $H(+\infty)$ into p polygonal lines H_1, \dots, H_p , each consisting of α edges; we set $m = \alpha(p - 1) + r$, so one of the polygonal lines will have $r < \alpha$ edges. Each H_i can be regarded as an unbounded convex polygon by stretching its end-edges to infinity. This allows us to apply Lemma 1 with respect to each point p_{m+1}, \dots, p_n and each polygon H_1, \dots, H_p . The maximum of the set formed by t_2 and the $(n - m)p$ values thus obtained is exactly the threshold of $H(+\infty)$. We easily see that the time complexity of the algorithm is in $O(p(n - m)4^d \log \alpha + p\alpha^{2^{2d+9}})$. Setting $\alpha = (n \log n)^{1/2^{2d+9}}$, we conclude

Theorem 3. In $O(n^{2^{-1/2^{2d+9}}} (\log n)^{1-1/2^{2d+9}}) = O(n^{2^{-1/4^{d+5}}})$ time, it is possible to compute the threshold of n points moving according to a polynomial function of time of degree d .

5. Conclusions

The main contribution of this work has been to show that Dobkin and Lipton's method [DL] for searching among hyperplanes can be generalized to handle arbitrary algebraic varieties. Our method is an adaptation of a quantifier-elimination procedure due to Collins [C]. This feature gives even more generality to our algorithm than mentioned earlier. Indeed, we do not have to limit ourselves to real algebraic varieties but may consider the more general problem of discriminating among *semi-algebraic sets*. Recall that a set $S \subseteq E^r$ is semi-algebraic if there exists a first-order sentence $\phi(x_1, \dots, x_r)$ in the theory of real numbers, with x_1, \dots, x_r as the only free variables of ϕ , such that $S = \{x \in E^r \mid \phi(x) \text{ is true}\}$. The preprocessing involves eliminating each quantifier by means of Collins projections (one projection per quantifier), and the point location takes place in the induced cylindrical algebraic decomposition of E^r . A similar technique was implicitly used in eliminating the time variable in the preprocessing of Section 4.

Further work includes the (difficult) problem of drastically reducing the high space-complexity of the generalized point location algorithm. Even the case of hyperplanes is still open. Regarding the problem of computing thresholds in steady-state computations, one will observe that our technique is general enough to be applied to other problems (e.g. closest/farthest pairs). An interesting question is to determine whether ad hoc treatment of these problems leads to more efficient solutions and thus our technique is in a sense too general, or if Theorem 3 is essentially all we can hope for. For example, a continuity argument easily shows that ensuring the local coherence of the steady-state Voronoi diagram is sufficient to compute its threshold (e.g. checking the non-zero length of its edges). It is then fairly simple to devise an $O(n \log n)$ algorithm for computing the steady-state Voronoi diagram of n moving points as well as its threshold. Note that the same argument can be made for convex hulls if all the points are guaranteed to lie on it. One essential feature of these easy cases is that the output *involves* all the input. Is this in general a necessary condition of efficiency?

Acknowledgments: Thanks to Prof. Thomas Banchoff for a stimulating discussion on algebraic topology. Thanks to Janet Incerpi for providing useful comments about this manuscript.

REFERENCES

- [A] Atallah, M.J. *Dynamic computational geometry*, 24th Annual FOCS Symp., pp. 92-99, Nov. 1983.
- [BT] Brown, W., Traub, J.F. *On Euclid's algorithm and the theory of subresultants*, J. ACM 18, pp. 505-514, 1971.
- [C] Collins, G.E., *Quantifier elimination for real closed fields by cylindric algebraic decomposition*, Proc. 2nd GI Conf. on Automata Theory and Formal Languages, Springer-Verlag, LNCS 35, Berlin, pp. 134-183, 1975
- [DL] Dobkin, D.P., Lipton, R.J. *Multidimensional searching problems*, SIAM J. Comput. 5(2), pp. 181-186, 1976.
- [E] Edelsbrunner, H., Guibas, L., Stolfi, J. *Optimal point location in a monotone subdivision*, to appear.
- [O] Overmars, M.H. *The locus approach*, Tech. Rept. RUU-CS-83-12, Univ. Utrecht, July 1983.
- [SS] Schwartz, J.T., Sharir, M. *On the "piano movers" problem. II: General techniques for computing topological properties of real algebraic manifolds*, Adv. in Appl. Math 4, pp. 298-351, 1983.
- [T] Tarski, A. *A decision method for elementary algebra and geometry*, Univ. of Calif. Press, 1948, 2nd edition, 1951.
- [W] van der Waerden, B.L. *Modern Algebra*, Ungar Co., New York, 1953.
- [Y] Yao, A.C. *On constructing minimum spanning tree in k -dimensional space and related problems*, SIAM J. Comput. 11(4), pp. 721-736, 1982.