Computing Roadmap

In this paper we study the problem of concerned with sets $S \subseteq R^n$, our algorithm computes a one-dimensional embedding of $S$ in a space $R^d$ for $R$. $R(S)$ is called the roadmap of the algorithm described in [Can88a], [Can89]. We measure the complexity of the polynomials $P$, their maximum degrees $d$, and the number of variables $n$. The complexity of our new algorithm is $\Theta(n^2)$ deterministic. Note that the combinatorial complexity of both cases is within a polylog factor.

1 Introduction

Recently, attention has turned to computing the number of connected components of semi-algebraic sets. This was described by Schwartz and Sharir [SS]. Their algorithm was designed for deciding adjacency between cells, and for the cell decomposition [BOKR96] (this is a much more difficult problem). Their algorithm was designed for the case where the input set is semi-algebraic and the cells are regular.

\*Supported by a David and Lucile Packard Foundation Grant IR-8858377
the form $B(A_1, \ldots, A_k)$ where $B : \{0, 1\}^k \rightarrow \{0, 1\}$ is a boolean function and each $A_i$ is an atomic formula of one of the following types:

$$(F_i = 0) \quad (F_i \neq 0) \quad (F_i > 0) \quad (F_i < 0) \quad (F_i \geq 0) \quad (F_i \leq 0)$$

with $F_i \in \mathbb{Q}[e]$. In the analysis that follows it will be helpful to assume a certain form for the defining predicate.

**Definition** A formula $B(A_1, \ldots, A_k)$ is said to be in monotone standard form if the boolean function $B$ is monotone, and all atomic formulae $A_i$ are either $(F_i = 0)$ or $(F_i > 0)$.

An arbitrary formula can be converted to monotone standard form with a constant factor increase in size. Assume we are given a boolean circuit $C$ to represent the function $B$. This circuit can be converted to a negation-free, and therefore monotone circuit $C_M$ as follows. For each node $v \in C$ there are two in $C_M$, one of which represents $v$ and the other $\overline{v}$. Now all primitive logical operations between nodes in $C_M$ can be implemented with $\land$ and $\lor$, e.g. if $C_T = C_3 \lor \overline{C_3}$ in the circuit $C$, in $C_M$, the node representing $C_T$ is the $\lor$ of the node representing $C_3$ and the node representing $\overline{C_3}$. We also need to compute $\overline{C_T}$ for later use, and this node is the $\land$ of the node representing $\overline{C_3}$ and the node representing $C_3$.

The circuit $C_M$ defines a monotone boolean function of the original atomic formulae and their negations. The negations can be pushed into the atomic formulae by replacing $(F_i > 0)$ with $(F_i \leq 0)$ etc. This formula can be converted to standard form by substituting for the inequalities $\leq, \not\equiv$ with a union of a pair of inequalities using $>$ and $\equiv$. Overall the number of atomic formulae increases by a factor of at most four compared to the original predicate.

**Predicate complexity** We measure the complexity of a predicate with four quantities, the number of polynomials $k$, the number of variables $n$, the maximum degree of the polynomials $d$, and the maximum coefficient length $c$ of the coefficients of the polynomials.

There remains one sticky point with regard to the boolean formula $B$. There is a certain cost associated with evaluating $B$, given the signs of the $F_i$'s. This time is clearly linear for the first evaluation. The algorithm of [Can88a] requires frequent re-evaluation of $B$ when a single $F_i$ changes sign. Our complexity bounds will be valid if the time to re-evaluate $B$ when a single $F_i$ changes sign is $O(\log k)$. In [Can90] we show that this will be true if the function $B$ is defined by a formula. Basically, we show there that an arbitrary boolean formula can be converted to an equivalent log-depth, fan-in two formula (in polynomial time). The predicate $B$ has been assumed to be a formula in most previous work on semi-algebraic sets. Our algorithm will still work if the function $B$ is represented by a general circuit, but if the time to re-evaluate $B$ when a single input changes is greater than $\log k$, then we must substitute this larger time for $\log k$ in the complexity bounds above.

**2.1 Stratifications**

**Definition** A stratification $\mathcal{S}$ of a set $S \subset \mathbb{R}^n$ is a partition of $S$ into a finite number of disjoint subsets $S_i$, called strata such that each $S_i$ is a manifold.
Lemma 3.1 There exists a positive \( r_0 \) such that for all \( r > r_0 \), \( S \cap D_r \) is a deformation retract of \( S \).

Proof Let \( F = (F_1, \ldots, F_k) \) be a set of polynomials which define \( S \). Let \( \mathbb{R}^n \) be a Whitney regular stratification of \( \mathbb{R}^n \) which is compatible with the sign-invariant sets of \( F \). All sign-invariant sets are unions of strata. Such a stratification is described in [BCR87]. By the semi-algebraic Sard theorem [BCR87], the map \( \rho \) has only finitely many critical values when restricted to any of these strata. Let \( r_0 \) be the largest critical value, then for all \( r > r_0 \), \( S \cap D_r \) has the same homotopy type (hence number of connected components) as \( S \). To see this, use \( \rho \) to lift a vector field on \([r, \infty)\) to one on \( \mathbb{R}^n - \text{Int}(D_r) \) which is is compatible with the stratification of \( \mathbb{R}^n \). The flow along this vector field defines a retraction of \( S \) onto \( S \cap D_r \).

So to find connected components of \( S \) it suffices to find components of \( S \cap D_r \) for sufficiently large \( r \). In practice this is done by treating \( r \) as an indeterminate element of the base field. When it comes time to determine the sign of a base field element, which will be a polynomial in \( r \), we use the sign of the highest degree term in \( r \). This correctly gives the sign for sufficiently large \( r \).

4 Reduction to a regular stratification

As was shown in [Can88a] one can obtain a regular stratification by taking the sign-invariant sets of a system of polynomials in sufficiently general position. In the present case, the given \( F \) will not be in general position. In [Can88a] a fixed perturbation was applied to their constant coefficients to achieve this. Now consider the following symbolic perturbation of the \( F = (F_1, \ldots, F_k) : \mathbb{R}^n \to \mathbb{R}^n \). Define

\[
F_{\alpha} = F + \alpha a
\]

where \( a \in \mathbb{R}^n \) is supposed constant for the time being, and \( \alpha \) is a single infinitesimal. As shown in [Can88a] for almost all choices of the constant coefficients \( \alpha a \), the map \( F_{\alpha} \) is transversal to the natural stratification by sign \( (\mathbb{R})^k \) of \( \mathbb{R}^n \), where \( \mathbb{R} = (\mathbb{R}^\ast, (0), \mathbb{R}^\ast) \).

In particular, if we fix \( \alpha = (\alpha_1, \ldots, \alpha_k) \), then for almost all such choices, the map \( F_{\alpha} \) is transversal to \( (\mathbb{R})^k \) for almost all \( \alpha \). This implies that the sign-invariant sets of \( F_{\alpha} \) form a Whitney regular stratification, a very important property for us later. We assume for now that such \( \alpha \)'s have been found. Later we will show how they can be determined either deterministically or probabilistically.

Assume the \( \alpha \)'s were chosen to be positive, the sign-invariant sets of (2) are the same as for

\[
F_1/\alpha_1 + \epsilon \\
F_2/\alpha_2 + \epsilon \\
\vdots
\]

Define \( G_{\alpha} = F_i/\alpha_i \). Then another way to describe the sign-invariant stratification of the last paragraph is as the preimage under \( G \) of the stratification \( (\mathbb{R})^k \).
Lemma 4.3 There exists a positive $\delta_0$ such that for all positive $\delta < \delta_0$ deformation retract of $F_\delta$.

Proof: Consider the set $D$ in $\mathbb{R}^{n+1}$ defined as $D = \{ (x_1, \ldots, x_n, \delta) \mid D(x_1, \ldots, x_n, \delta) \}$ of $D$ as the "graph" of $F_{\delta}$. $D$ is certainly semi-algebraic, and so its stratification compatible with the signs of the polynomials $F_\delta$ and $(F_\delta, x_1 : (x, \delta) \mapsto \delta)$ has a finite number of critical values when restricted to $\partial D$ to be the smallest positive critical value, and let $\delta$ be any positive $\delta$. Then $x_1$ is regular on all strata for the range of the values $(0, \delta]$. If a vector field on $(0, \delta]$ to a vector field on $D$ which is compatible with $D$ is compact when restricted to $[0, \delta]$, this gives us a deformation $\psi^\delta(0, \delta)$ onto the compact set $D_0 = \psi^\delta(0, \delta)$. But if we define $\pi_\delta : (x, \delta) \mapsto x$, then the projection $\pi_\delta(D(0, \delta)) = D_0$ is just $F_{\delta}$. Furthermore, composing $\pi_\delta$ with the deformation retraction paragraph gives us a deformation retraction of $F_\delta$ onto $F_{\delta}$. 

To guarantee that $\delta$ is small enough, we leave $\delta$ as an indeterminate field (like $r$), and when it comes time to evaluate the sign of a base for a polynomial in $r$ and $\delta$, we first find the term of lowest degree in $\delta$, with this degree in $\delta$, we take the sign of the highest degree term in $r$. Proceeding all evaluations with the quantification $\exists r \forall \delta \exists \delta_0 \forall \delta < \delta_0$.

Operationally, this is also equivalent to working in a real field extent larger than any element of $\mathbb{R}$ and $\delta$ is smaller than any element of the other. However, in the algorithms that follow, all the numerical calculations polynomials from $\mathbb{R}[x, \delta]$. We do not need $r$ and $\delta$ to be values that only "sufficiently large" or "sufficiently small" real values that the signs are correctly computed. If we work only over the reals, life is much easier of compactness and connectivity apply.

In the last lemma we defined compact sets $F_{\delta}$ with the useful are compact and are deformation retracts of sign-invariant sets. The connected components of a sign-invariant set $F_\delta$ contains a single component we define a set $F_{\delta}'(\delta, \epsilon)$ which is a "neighborhood" of $F_{\delta}(\delta, \epsilon)$ and contain it. To define $F_{\delta}'(\delta, \epsilon)$, we take each inequality in an $F_\delta$ and replace it by $\delta$.

\begin{align*}
\delta + \epsilon & \geq 0 \\
\delta & \geq 0 \\
\delta & \geq 0 \\
\delta & \geq 0
\end{align*}

and notice that $F_{\delta}'(\delta, \epsilon)$ is now compact. By the previous general proper, regularly stratified by the signs of the polynomials define it. Now $F_\delta$ is a subset of $F_{\delta}'(\delta, \epsilon)$.

Lemma 4.4 Assume $\delta$ is chosen to satisfy lemma 4.3. There exists a for all $0 < \epsilon < \epsilon_0$, the set $F_{\delta}'(\delta, \epsilon)$ is a deformation retract of $F_{\delta}'(\delta, \epsilon)$.

Proof: We only sketch this proof since it is almost identical to the other. Let $D$ in $\mathbb{R}^{n+1}$ defined as $D = \{ (x_1, \ldots, x_n, \delta) \mid x \in F_{\delta}'(\delta, \epsilon) \}$. 

\begin{align*}
&\delta + \epsilon \\
&\delta \\
&\delta \\
&\delta
\end{align*}
For all sufficiently small $\varepsilon > 0$, the set $D$ has a Whitney regular stratification $\mathcal{D}$ into sign-invariant sets of the polynomials that define it. There is some $\delta_0$ which is the smallest positive critical value of the projection $\pi : (x, \varepsilon) \mapsto \varepsilon$ restricted to $D$. For $\varepsilon < \delta_0$, we can use $\pi$ to lift a vector field on $(0, \varepsilon)$ and thereby define the desired retraction. \[ \nabla \]

Notice that $F^+(\delta, \varepsilon)$ is exactly the set $G^{-1}(\varepsilon_{0,0})$, the non-singular approximation of $F^-(\varepsilon)$. Since $F_2$ and $F_2^+(\delta, \varepsilon)$ have a common retract $F_2^+(0, \varepsilon)$, they have the same homotopy type and hence the same number of connected components. This completes the proof of theorem 4.2.

Finally, we observe that the sets $F^+(\delta, \varepsilon)$ and $F_2^+(\delta, \varepsilon)$ are disjoint for $\varepsilon$ and $\varepsilon'$ distinct. This follows because if the union of a component of $F_2^+(\delta, \varepsilon)$ and a component of $F^+(\delta, \varepsilon)$ were connected, then its image under $G$ would have to be connected also. But that image must be in the union of the disjoint sets $\epsilon_{0,0}$ and $\epsilon'_{0,0}$, and it must intersect both, which is impossible.

So the connected components of $G^{-1}(\epsilon_{0,0})$, which are the union of connected components of $F_2^+(\delta, \varepsilon)$, correspond exactly to the connected components of sign-invariant sets of $F$. Since $G^{-1}(\epsilon_{0,0})$ is a compact set which is regularly stratified as $G^{-1}(\epsilon_{0,0})$, we can apply the algorithm of [Can88a], modified to work over arbitrary real coefficients fields as described in [Can88c].

5 Determining adjacencies between sign components

In the last section, we modeled connected components of sign-invariant sets $F_2$ with regularly stratified compact "neighborhood" sets $F_2^+(\delta, \varepsilon)$. In this section, we show how these neighborhood sets can also be used to determine adjacencies between components. We will need to make use of a "big" neighborhood of one set and a "small" neighborhood of the other.

First we remark that if disjoint sets $A$ and $B$ have a connected union, then either $A \cap B \neq \emptyset$, or $B \cap A \neq \emptyset$. For if these were not true, both $A$ and $B$ would be open in the union $A \cup B$, and therefore both closed in $A \cup B$. So to check whether connected $A$ and $B$ have a connected union, it suffices to check whether either $A \cap B \neq \emptyset$, or $B \cap A \neq \emptyset$.

Let $A$ be a connected component of $F_2$, and let $A^+(\delta, \varepsilon)$ be the corresponding connected component of $F_2^+(\delta, \varepsilon)$. Let $B$ be a connected component of a second sign-invariant set $F_2$, and $B^+(\delta_0, \varepsilon_0)$ be the corresponding connected component of $F_2^+(\delta_0, \varepsilon_0)$. Note the use of two different sets of infintesimalis. The next lemma shows that for suitable choices of the infinitesimals, adjacency of $A$ and $B$ can be checked by testing overlap of $A^+(\delta, \varepsilon)$ and $B^+(\delta_0, \varepsilon_0)$:

Lemma 5.1 With $A^+(\delta, \varepsilon)$ and $B^+(\delta_0, \varepsilon_0)$ as defined above, and for all sufficiently small $\delta \gg \varepsilon$ and $\delta_0 > \varepsilon > 0$,

$$A \cap B \neq \emptyset \iff A^+(\delta, \varepsilon) \cap B^+(\delta_0, \varepsilon_0) \neq \emptyset$$

Proof First we clarify what we mean by "sufficiently small". Let $\varepsilon_0$ mean that there is some positive $\varepsilon'$ such that the formula that follows the quantifier is true for all positive $\varepsilon$ less than $\varepsilon'$. The lemma states that,

$$\forall \delta \exists \varepsilon_0 \exists \xi_0 \exists \xi_0 (\varepsilon_0) \exists \xi_0 \exists \xi_0 (\varepsilon_0) \exists \xi_0$$

We can simplify this formula by doing $B^+(\delta_0)$ be the component of $F_2^+(\delta_0)$:

$$\forall \varepsilon_0 \forall (A^+(\delta_1, \varepsilon_1) \cap B^+(\delta_0, \varepsilon_0) \neq \emptyset$$

To see this, notice that the set $(A^+(\delta_1, \varepsilon_1) \cap B^+(\delta_0, \varepsilon_0))$ is a point in $(A^+(\delta_1, \varepsilon_1) \cap B^+(\delta_0, \varepsilon_0))$, and $\delta_1$ excluded those $F_2^+$ which are zero at $\delta_0$. The first formula will be true for all $\delta_0$ if any $\delta_0$, there is then a point $p$ in $A^+(\delta_1, \varepsilon_1)$ since $B^+(\delta_0)$ is a subset of $B$, so the we

$$\forall \delta_0 A^+(\delta_1, \varepsilon_1) \cap B^+(\delta_0, \varepsilon_0)$$

Suppose first that $A \cap B \neq \emptyset$, and let $p$ be in the (relative) interior of $A^+(\delta_1, \varepsilon_1)$, arguments. We know that all neighbor checks this. Specifically, for every $\varepsilon_1$, $U$ of $p$. This follows because all the $\delta_1$ are in the (relative) interior of $A^+(\delta_1, \varepsilon_1)$, $U$. Conversely, suppose $A \cap B = \emptyset$. We $\delta_1 > 0$. Pick a $\delta_1 > 0$. Now choose positive $\varepsilon_1$ values. Then for least on $\delta_1$, $\delta_1 > 0$. Why? Suppose the intersection $A^+(\delta_1, \varepsilon_1) \cap B$. The sequence $(p_\delta)$ lies convergent subsequence. But let $p$ be continuity of the polynomials defining converges to $p \in A$. This shows $A \cap B$.

To compute with this quantificational elements of the ground field. This is $R(\delta_1, \varepsilon_1, \delta_0, \varepsilon_0)$ where $\delta_1$, $\varepsilon_1$, $\delta_0$ and $\varepsilon_0$ smaller than any positive element of.
A certain compact, real line $\mathbb{R}_0$ which consists between and ido do.

Let $S$ be the ori without loss of gener function of inequality chosen so that $G = \text{formula for } S$ wi
and each $(F_i > 0)$ will formula to be monoton

Similarly, we can result relating to $S_{\text{opt}}$.

**Corollary 5.2** There and connected components.

**Proof** Let $A_1, \ldots$ that are contained in same sign-invariant s, others must be.

We saw in the last connected components. Now consider the $u_i$ intersects $A_i$. We can and $A_i$ if and only if $A_i \cup A_j$ is contained in $A_i \cap A_j = A_i^*(\delta_i, \epsilon_i) \cap A_j^*(\delta_j, \epsilon_j)$

Conversely, suppose $A_i^*(\delta_i, \epsilon_i)$ and $A_j^*(\delta_j, \epsilon_j)$ two remaining posit $A_i^*(\delta_i, \epsilon_i) \cap A_j^*(\delta_j, \epsilon_j)$

5.1 Transformations

To summarize, the following general semi-algebra...
3. Each query polynomial contains at most $O(n^3)$ infinitesimals, even if all of $a_1, \ldots, a_q$ are infinitesimal. This seems surprising at first, but one must remember that by using infinitesimal $a_i$'s, we have guaranteed that all the algebraic surfaces defined by the input polynomials meet transversally. In particular, any collection of more than $n$ surfaces will not have a common intersection point. If the query polynomials were only generated by intersection points, there could be at most $O(n)$ infinitesimals in each one. But the roadmap algorithm also generates hyperplanes which are defined by $n$ input surfaces. These also meet the surfaces transversally (except at one point each), and so an intersection point can actually depend on $O(n^2)$ input surfaces.

Making these observations allows us to determine the running time for the deterministic version of the algorithm, which uses infinitesimal $a_i$'s. The bounds in [Can88a] and [Can88c] show that the roadmap algorithm for a particular input requires $O(k \log k)O(n^5)$ evaluations of query polynomials. The above arguments indicate that each polynomial has degree $n^{O(n)}$ in $O(n^5)$ variables, or $n^{O(n)}$ coefficients, and they require $O(n^3)$ time to evaluate (coefficient length bounds are much better than this, and don't affect the complexity bounds). The running time is the product of the number of query polynomial evaluations and the time for each of which is $(k \log k)O(n^5)$.

For the randomized algorithm, we need to figure out the number of random bits required in the choice of $a_i$'s. We could try to figure out explicitly the conditions for a particular $a$ to be a good choice, but there is a simpler argument we can use, which takes advantage of the fact that our calculation can be expressed as an algebraic decision tree. A particular $a$ must be a good choice if all the query polynomials in the decision tree are non-zero at that $a$ (excepting query polynomials which are identically zero, which can be ignored). This follows because for each such an $a$, there is an open, connected neighborhood $N(a)$ such that all the query polynomials have the same sign over all of $N(a)$ as they do at $a$. Thus the algorithm's output is the same for all these choices. But almost all of the points in $N(a)$ must be good choices, since good points are dense. The algorithm must produce the correct output at these points, hence it produces the correct output at $a$.

So it suffices to choose $a$ to avoid the zero sets of all the query polynomials. The query polynomials have degree $n^{O(n)}$ and there are potentially $(k \log n)O(n^5)$. The union of the zero sets gives us a bad set which is an algebraic set in the space $\mathbb{A}^n$ of possible $a$-values which has degree $(k \log n)O(n^5)$. By Schwartz's lemma, we will have probability $p$ of hitting the bad set if we choose the $a$'s randomly with $\log(p) = (k \log n)O(n^5)$ bits. Fixing $p$, we see that $O(n^3 \log(k))$ bits suffice for the $a$'s. This contributes the extra factor of $\log k$ to the running time of the original roadmap algorithm, and gives a randomized running time of $(k \log k)O(n^5)$.\end{proof}

References


