

Research Report

A NOTE ON SENSITIVITY ANALYSIS IN ALGEBRAIC ALGORITHMS

Nimrod Megiddo

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099,
Mathematical Sciences Research Institute, Berkeley, California
and Tel Aviv University, Tel Aviv, Israel

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties).



Research Division
Yorktown Heights, New York * San Jose, California * Zurich, Switzerland

A NOTE ON SENSITIVITY ANALYSIS IN ALGEBRAIC ALGORITHMS

Nimrod Megiddo

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099,
Mathematical Sciences Research Institute, Berkeley, California
and Tel Aviv University, Tel Aviv, Israel

ABSTRACT: This note discusses a problem that is analyzed in a recent report by Eaves and Rothblum. It deals with computation over any ordered field F . However, knowledge of ordered field theory is not required. The model of computation is that of the random access machine where each register holds a single *element* $\alpha \in F$. The primitive operations are the arithmetic operations $+$, $-$, \times and \div , comparisons and data transfers. It is shown how algorithms can be extended to solve problems with parametric data.

A Note on Sensitivity Analysis in Algebraic Algorithms

Nimrod Megiddo

We discuss here one problem that is analyzed in a recent report by Eaves and Rothblum [ER]. Previous papers that used related methods are listed in the references. Our discussion here is about computation over any ordered field F . However, knowledge of ordered field theory is not required here. For the reader who is interested in concrete computation, the ordered field of the rational numbers may be taken as an example. We do not need F to be real closed or archimedean. The model of computation is the random access machine where each register holds a single *element* $\alpha \in F$.

The primitive operations are the arithmetic operations $+$, $-$, \times and \div , comparisons (that is, given two elements $\alpha, \beta \in F$, the machine can recognize whether $\alpha < \beta$) and, of course, data transfers. In this model an algorithm \mathcal{A} receives input in the form of a vector $\alpha = (\alpha_1, \dots, \alpha_n) \in F^n$. The algorithm outputs a vector $\beta = (\beta_1, \dots, \beta_m) \in F^m$. We note that the elements 0 and 1 can be used in the usual way to encode information so that this model can in particular be used to solve combinatorial problems with data taken from F . It is also obvious that in this model an algorithm can, in particular, recognize whether a given element α equals 0 or 1, even if the elements 0 and 1 are not given explicitly. Without further restrictions an algorithm may attempt to divide by 0 and also may not halt.

Definition 0.1. Let \mathcal{A} be an algorithm. An input $\alpha \in F^n$ is *valid* for \mathcal{A} if, given α , \mathcal{A} reaches the halting state. In particular, throughout the execution \mathcal{A} never attempts to divide by 0.

We are interested in the following question of parametric extension of algorithms in this model. Imagine that the input vector $\alpha = (\alpha_1, \dots, \alpha_n)$ consists of the values of n rational functions (over F) $f_1(t), \dots, f_n(t)$ (for example, $\alpha_i = f_i(0)$). Furthermore, suppose there is some $\epsilon > 0$ such that $\alpha = f(t)$ is a valid input for \mathcal{A} for any t such that $|t| < \epsilon$. We would like to extend \mathcal{A} into an algorithm \mathcal{A}^* that computes some $\delta > 0$ and provides a “closed-form” description of the output of \mathcal{A} relative to *all* t such that $|t| < \delta$. This is the problem of *sensitivity analysis* or local parametric computing. The sense of the closed-form is explained below. Essentially, this means a piecewise rational function with a small number of pieces.

We will describe a fairly simple extension of this kind that is valid under a natural assumption as follows. Given the vector-function f , let $T_f(t)$ denote the running time (that is, the number of primitive operations executed by \mathcal{A}) in case the algorithm is given the values of f at t (assuming $|t| < \epsilon$). The natural assumption is that $T_f(t)$ is *bounded over some neighborhood of 0*.

The idea behind the extension is fairly intuitive. Rational functions can be operated with like field elements. Of course, the costs of the operations (that is, the number of elementary operations

of the machine) depend on the degrees of the polynomials involved and the specific algorithms used for carrying out these operations on rational functions. Suppose a rational function $g(t) = p(t)/q(t)$ is represented by two vectors of field elements containing the coefficients of the polynomials $p(t)$ and $q(t)$. Then it is straightforward to carry out and represent the results of the arithmetic operations applied to rational functions. If one is interested in maintaining an *interval* of values of t over which all the current functions are well-defined, then one has to exclude zeros of functions $h(t)$ before executing a division $g(t) \div h(t)$. We explain below how this can be done. Comparisons between two functions depend of course on the value of t . However, when we compare two *rational* functions $g(t)$ and $h(t)$, well-defined in a neighborhood of 0, there is a $\delta > 0$ such that the comparison is uniform over each of the open intervals $(-\delta, 0)$ and $(0, \delta)$. In other words, either $g(t) < h(t)$ for all t in the same interval, or $g(t) \geq h(t)$ for all t in the same interval. We show below how to determine such a δ . It follows that we can extend an algorithm \mathcal{A} so that the “program variables” become rational functions rather than field elements. More precisely, at any instant during the execution there is a $\delta > 0$ such that for each program variable X there are at most two more “variables” associated with X : (i) a rational function $X_+(t)$ well-defined over $(0, \delta)$ and (ii) a rational function $X_-(t)$ well-defined over $(-\delta, 0)$. Often there will be just one function $X(t)$ from which $X_-(t)$, X and $X_+(t)$ are obtained trivially by restriction. To simplify the discussion, let us without loss of generality consider only the interval $(0, \delta)$.

There is an easy estimation problem that needs to be solved each time the extended algorithm attempts to carry out a division or a comparison. Suppose $g(t) = p(t)/q(t)$ and $h(t) = r(t)/s(t)$ are rational functions, that is, p, q, r and s are polynomials such that $q(t)$ and $s(t)$ do not vanish in $(0, \delta)$. We wish to find a δ' , $0 < \delta' \leq \delta$ such that $i(t) = g(t)/h(t)$ is well-defined over $(0, \delta')$. The representation of $i(t)$ as a quotient of two polynomials is easily obtained by polynomial multiplication. For comparison we have to determine a δ' such that either $g(t) < h(t)$ for all $t \in (0, \delta')$ or $g(t) \geq h(t)$ for all $t \in (0, \delta')$. In both cases the problem of determining δ' can obviously be reduced to the following problem:

Given a polynomial $p(t) = \pi_0 + \pi_1 t + \dots + \pi_n t^n$ (assuming $\pi_n \neq 0$), find a $\lambda > 0$ such that $p(t) \neq 0$ for all $t \in (0, \lambda)$.

The solution of this problem is summarized in the following proposition.

Proposition 0.2. *Let $p(t) = \pi_0 + \pi_1 t + \dots + \pi_n t^n$ ($n \geq 1$) be a polynomial over F such that both $\pi_n \neq 0$ and $\pi_0 \neq 0$. Let $\lambda = |t| < \text{Min}\{1, |\pi_0| / (|\pi_1| + \dots + |\pi_n|)\}$. Under these conditions, for every t such that $|t| < \lambda$, $p(t) \neq 0$.*

Proof. Suppose t satisfies the condition stated in the proposition. Note that the triangle inequality $|\alpha + \beta| \leq |\alpha| + |\beta|$ holds over any ordered field. It follows that

$$|p(t)| \geq |\pi_0| - (|\pi_1| |t| + \dots + |\pi_n| |t|^n)$$

$$> |\pi_0| - (|\pi_1| + \dots + |\pi_n|) |t| > 0.$$

Obviously, the assumption that $\pi_0 \neq 0$ is not restrictive since a factor of the form t^j can be eliminated.

This discussion leads to the following theorem:

Theorem 0.3. Suppose \mathcal{A} is an algorithm over F and $f(t) = (f_1(t), \dots, f_n(t))$ is a vector of rational functions over F satisfying the following conditions:

- (i) There exists an element $\epsilon > 0$ such that for every $t \in (-\epsilon, \epsilon)$, $f(t)$ is a valid input for \mathcal{A} .
- (ii) There exists an integer $\bar{T} = \bar{T}(f)$ such that for every $t \in (-\epsilon, \epsilon)$, \mathcal{A} halts on $f(t)$ within \bar{T} elementary operations.

Under these conditions, there exists an element δ , $0 < \delta \leq \epsilon$, such that over each of the open intervals $(-\delta, 0)$ and $(0, \delta)$, \mathcal{A} executes the same sequence of elementary operations for all t in the same interval. Furthermore, the outputs can be described by rational functions over these intervals, and these functions can be computed by an extended algorithm \mathcal{A}^* within $\bar{T}(f)$ "elementary" operations on rational functions.

Proof. The proof of the theorem goes by induction on the number k of the step in the execution of the algorithm. Obviously, the extended algorithm simulates the original algorithm over two nested sequences of intervals $(-\delta_k, 0)$ and $(0, \delta_k)$. There remains the question of halting. But, since \mathcal{A} is assumed to obey a uniform time bound in a some neighborhood of 0 (given the rational functions as input) it follows that \mathcal{A}^* halts within this uniform time bound, interpreted as the number of rational functions operations.

Note that in numbers of elementary field operations \mathcal{A}^* runs longer, and obviously the degrees of the polynomials involved may grow exponentially during the execution. Thus, usually, the running times of the two algorithms are not polynomially related. However, in the worst case, if \mathcal{A} terminates in \bar{T} operations over $(-\epsilon, \epsilon)$ then \mathcal{A}^* requires no more that $2^{O(\bar{T})}$ operations. Also, note that Tarski's principle was not used in our construction.

References

- [A] M. J. Atallah, "Dynamic computational geometry", in: *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (1983)*, IEEE Computer Society Press, Los Angeles, 1983, pp. 92-99.
- [Co] R. Cole, "Slowing down sorting networks to obtain faster sorting algorithms" in: *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science (1984)*, IEEE Computer Society Press, Los Angeles, 1984, pp. 255-260.

- [CSY] R. Cole, M. Sharir and C. Yap, "On k -hulls and related problems" in: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing (1984)*, ACM, New York, 1984, pp. 154-166.
- [ER] B. C. Eaves and U. G. Rothblum, "A theory of extending algorithms for parametric problems", Technical Report, August 1985.
- [G] D. Gusfield, "Parametric combinatorial computing and a problem of program module distribution", *Journal of the Association for Computing Machinery* **30** (1983), pp. 551-563.
- [J1] R. G. Jeroslow, "Asymptotic linear programming", *Operations Research* **21** (1973) pp. 1128-1141.
- [M1] N. Megiddo, "Combinatorial optimization with rational objective functions", *Mathematics of Operations Research* **4** (1979) pp. 414-424.
- [M2] N. Megiddo, "Towards a genuinely polynomial algorithm for linear programming", *SIAM Journal on Computing* **12** (1983) No. 2, pp. 347-353.
- [M3] N. Megiddo, "Applying parallel computation algorithms in the design of serial algorithms", *Journal of the Association for Computing Machinery* **30** (1983) No. 4, pp. 337-341.
- [M4] N. Megiddo, "The weighted Euclidean 1-center problem", *Mathematics of Operations Research* **8** (1983) No. 4, pp. 498-504.
- [M5] N. Megiddo, "Dynamic location problems", *Annals of Operations Research* **5** ("Location decisions: methodology and applications"), to appear.
- [MT1] N. Megiddo and A. Tamir, "Finding least-distances lines", *SIAM Journal on Algebraic and Discrete Methods* **4** (1983) No. 2, pp. 207-211.
- [MT2] N. Megiddo and A. Tamir, "New results on the complexity of p -center problems", *SIAM Journal on Computing* **12** (1983) No. 4, pp. 751-758.
- [MZ] N. Megiddo and E. Zemel, "A randomizing $O(n \log n)$ algorithm for the weighted 1-center problem", *Journal of Algorithms*, to appear.