JOURNAL OF COMPLEXITY 8, 393–397 (1992)

A Problem That Is Easier to Solve on the Unit-Cost Algebraic RAM

PRASOON TIWARI*

Department of Computer Science, University of Wisconsin–Madison, 1210 W. Dayton Street, Madison, Wisconsin 53706

Received February 3, 1992

Problem 5.1 in (L. Blum, M. Shub, and S. Smale, 1989, *Bull. Amer. Math. Soc.* **21**(1), 1–46) asks if there is a *decision* problem that cannot be solved in polynomial time by a Turing machine, but can be solved in polynomial time on a unit-cost algebraic RAM with operations $\{+,-,*,/,<\}$, and without the integer division operation. We present a problem that is not known to be solvable in polynomial time on a Turing machine, but can be solved in polynomial time on a unit-cost algebraic RAM. This is strong evidence for an affirmative answer to Problem 5.1. © 1992 Academic Press, Inc.

1. INTRODUCTION

In a recent paper Blum, Shub, and Smale (1989) define a model of computation over an arbitrary ring, and study the notion of universal machines, partial recursive functions, and *NP*-Completeness in this model. They discuss several measures of time required by basic operations in their model, e.g., the unit-cost measure and the logarithmic-cost measure. (See, for example, Aho *et al.*, 1974.)

In the context of models that allow arithmetic operations from the set $\{+,-,*,/,<\}$, but do not allow the truncation or the integer division operations, they pose the following problem:

Problem 5.1 (*Blum et al., 1989*). Would the class of *decision* problems in P (polynomial time) over $\mathbf{Z}$ change if the cost function were changed to unit-cost (instead of log-cost)?

* Partially supported by NSF under Grant CCR-9024516.

0885-064X/92 $5.00

If either integer division or truncation is allowed as a unit-cost operation, then Bertoni *et al.* (1976) showed that any problem in *#P—SPACE* can be solved in polynomial time on an algebraic RAM.

Although Problem 5.1 refers to the unit-cost model, the size of an instance is defined to be the number of bits needed to represent it. Often, this is a desirable definition. See, for example, the definition of strongly polynomial time in (Grötshel *et al.*, 1988) and the lower bound arguments in (Mansour *et al.*, 1991).

We point out that the following well known problem is not known to be in P, but can solved in polynomial time in the unit-cost model. Thus, it provides strong evidence for an affirmative answer to Problem 5.1 of (Blum *et al.*, 1989).

Problem (The Sign Problem). Given n primes of m-bits each, and a set of integers (coefficients) $c_{b_1,b_2,\ldots,b_n}$, for $b_i \in \{0, 1\}$, determine if the following expression is greater than zero:

$$\mathbf{E} = \sum_{b_i \in \{0,1\}; i=1,2,\ldots,n} c_{b_1,b_2,\ldots,b_n} \sqrt{p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}}. \tag{1}$$

The status of this problem has been open since (at least) 1976, when Garey *et al.* (1976) mentioned a similar problem in the context of *NP*-Completeness of a class of geometric problems. It is well known that $\mathbf{E}$ equals zero if and only if all of the coefficients $c_{b_1,b_2,\ldots,b_n}$ are zero (van der Warden, 1970). Borodin *et al.* (1985) gave a polynomial-time algorithm for testing if a general expression in integers involving square-roots *equals* zero. Yet, the Sign Problem is not known to be in *NP*. However, it is known to be in *PSPACE* (Borodin, 1982). In contrast, using the algorithm of Section 2, this problem can be solved in polynomial-time in the Blum–Shub–Smale model.

2. An Algorithm for The Sign Problem

In this section, we present an algorithm for solving the Sign Problem in polynomial time on a unit-cost algebraic RAM. We begin by defining the size of $\mathbf{E}$.

Denote the size of $\mathbf{E}$ by $s(\mathbf{E})$ and define it to be

$$s(\mathbf{E}) = nm + \sum_{b_i \in \{0,1\}; i=1,2,\ldots,n} \lceil \log_2(|c_{b_1,b_2,\ldots,b_n}| + 1) \rceil. \tag{2}$$

Note that $|\mathbf{E}| \leq 2^{s(\mathbf{E})}$. Also note that representing an instance requires at least $s(\mathbf{E})$-bits.

The key observation is that if $\mathbf{E} \neq 0$ then $|\mathbf{E}| \geq 2^{-(s(\mathbf{E})+1)2^n}$. We will show that, if one approximates the square-roots in Eq. (1) using Newton itera-

tion with appropriate starting points then, after $mn + \log((s+1)2^n + s + 2)$ iterations $\mathbf{E}$ can be computed to enough precision in order to solve the Sign Problem.

We will need the following well known theorem:

THEOREM 1 (Householder, 1970). *If $h(x)$ is a polynomial of degree d with k bit integer coefficients, then all of its roots are less than 2^k in absolute value.*

COROLLARY 2. *If $h(x)$ is a polynomial of degree d with k bit integer coefficients, then all of its roots are greater than $1/2^k$ in absolute value.*

Proof. If α is a root of $h(x)$ then $1/\alpha$ is a root of $x^d h(1/x)$. ■

The following lemma is also known. Similar results were mentioned in (Garey *et al.*, 1976) and (Borodin, 1982). We include a proof for completeness.

LEMMA 3: *If $\mathbf{E} \neq 0$, then $|\mathbf{E}| \geq 2^{-(s(\mathbf{E})+1)2^n}$.*

Proof. Assume that $\mathbf{E} \neq 0$. Clearly, $\mathbf{E}$ is an algebraic number of degree at most 2^n. Let $\sigma_1 = \mathbf{E}, \sigma_2, \ldots, \sigma_l$ be the l conjugates of $\mathbf{E}$; $l \leq 2^n$. Each of these conjugates is represented by an expression like the one in Eq. (1), with the exception that the signs of some of the terms in the summation may be different. Therefore, $|\sigma_i| \leq 2^s$.

The minimal polynomial of $\mathbf{E}$, given by

$$q(x) = \prod_{i=1}^{l} (x - \sigma_i) = \sum_{i=0}^{l} q_i x^i, \tag{3}$$

is monic, with integer coefficients. Since each q_i is the sum of at most 2^l terms, each of which is a product of at most l σ_i's, we have

$$|q_i| < 2^{sl+l}. \tag{4}$$

Now, Corollary 2 implies that $\sigma_i > 2^{-sl-l}$. Therefore, $|\mathbf{E}| > 2^{-sl-l}$. ■

We also need an algorithm for computing the square-root; Newton iteration will do for this application. Recall that, the Newton iteration for computing the square-root of a positive integer a can be written as follows:

$$x_0 = a \geq 0$$

$$x_{i+1} = \frac{x_i^2 + a}{2x_i}$$

$$x_i \to \sqrt{a}.$$

Note the following well known facts about this Newton iteration:

1. $x_i > \sqrt{a}$ implies $x_{i+1} > \sqrt{a}$.
2. If $e_i = x_i^2 - a$, then $\varepsilon_i > 0$ and $e_{i+1} = e_i^2/4x_i^2$.
3. If e_g is a positive, then $e_i \leq e_0/4^i$.

Therefore, if a is an m-bit integer, then quadratic convergence begins after at most m initial iterations.

THEOREM 4. *The Sign Problem can be solved in polynomial time on a unit-cost algebraic RAM.*

Proof. Apply k (to be determined below) Newton-iterations of each of the square-roots in $\mathbf{E}$. Multiply the reuslting approximation to a square-root by the corresponding coefficient, add all these partial results, and denote the resulting value by $\mathbf{e}$. If there is an additive error ε in each approximation of the square-root, then $|\mathbf{E} - \mathbf{e}| \leq 2^s\varepsilon$. By Lemma 3, if $\mathbf{E} \geq 0$, then $\mathbf{E} > 2^{-(s+1)2^n}$. Pick ε such that $|\mathbf{E} - \mathbf{e}| \leq (1/4)\, 2^{-(s+1)2^n}$. Then, $\mathbf{E} > 0$ if and only if $\mathbf{e} > (1/2)\, 2^{-(s+1)2^n}$. But this condition on ε can be realized by choosing $k = mn + \lceil \log_2((s+1)2^n + s + 2) \rceil$. ■

3. CONCLUDING REMARKS

We have provided strong evidence for an affirmative answer to Problem 5.1 of Blum *et al.* (1989). In an attempt to answer the same problem, Shub (to appear) considered the problem of determining if

$$\prod_{i=1}^{k} a_i^{n_i} > \prod_{i=1}^{l} b_j^{m_j},$$

for a fixed k and l. But he was able to solve this restriction in polynomial time. Can the problem still be solved in polynomial time if k and l are allowed to be inputs?

ACKNOWLEDGMENTS

Thanks to Manuel Blum for focusing my attention on Problem 5.1 of (Blum *et al.* (1989), and to Mike Shub for simplifying the original proof.

REFERENCES

AHO, A. V., HOPCROFT, J. E., AND ULLMAN J. D. (1974), "The Design and Analysis of Computer Algorithms," Addison–Wesley, Reading, MA.

BERTONI, A., MAURI, G., AND SABADINI, N. A characterization of the class of functions computable in polynomial time on random access machines, Proceedings of the 13th Annual ACM Symposium on Theory of Computing, pp. 168–176, 1981.

BLUM, L., SHUB, M., AND SMALE, S. (1989), On a theory of computation and complexity over the real numbers: *NP*-completeness, recursive functions and universal machines, *Bull. Amer. Math. Soc.* **21**(1), 1–46.

BORODIN, A. (1982), Lecture Notes for CSC 2408F, University of Toronto.

BORODIN, A., FAGIN, R., HOPCROFT, J., AND TOMPA, M. (1985), Decreasing the nesting depth of expressions involving square roots, *J. Symbolic Comput.* **1**, 169–188.

GAREY, M. R., GRAHAM, R. L., AND JOHNSON, D. S. (1976), Some NP-complete geometric problems, *in* "Eighth Annual ACM Symposium on Theory of Computing Hershey, Pennsylvania, May 1976," pp. 10–22.

GAREY, M. R. AND JOHNSON, D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco.

GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. (1988), "Geometric Algorithms and Combinatorial Optimization," Springer-Verlag, Berlin.

HOUSEHOLDER, A. S. (1970), "The Numerical Treatment of Single Nonlinear Equation," McGraw-Hill, New York.

MANSOUR, Y. SCHIEBER, B., AND TIWARI, P. (1991), Lower bounds for integer greatest common divisor computations, *J. Assoc. Comput. Mach.* **38**(2), 453–471.

SHUB, M. Some remarks on Bezout's theorem and complexity theory, *in* "Proceedings of the Smalefest '90" (Hirsch, Marstden, and Shub, Ed.), to appear.

VAN DER WARDEN, B. L. (1970), "Algebra," Vol. 1, Fredrick Unger.

ZIPPLE, R. (1985), Simplification of expressions involving radicals, *J. Symbolic Comput.* **1**, 189–210.