Discrete Applied Mathematics 9 (1984) 117–123
North-Holland

POLYNOMIAL TESTING OF THE QUERY "IS $a^b \geq c^d$?" WITH APPLICATION TO FINDING A MINIMAL COST RELIABILITY RATIO SPANNING TREE

R. CHANDRASEKARAN
University of Texas at Dallas, P.O. Box 688, Richardson, TX 75080, USA

A. TAMIR
Statistics Department, Tel Aviv University, Tel Aviv 69978, Israel

Received 10 December 1982
Revised 30 September 1983

Given integers a, b, c, d, we present a polynomial algorithm for the query "is $a^b \geq c^d$?". The result is applied to yield a polynomial algorithm for the minimal cost reliability ratio spanning tree problem.

Introduction

Suppose that a network is given in which for each edge there is a probability of functioning. It is well known, (see [14]), that a maximally reliable spanning tree can be found by imitating the procedure for finding a minimum spanning tree. One way to view this approach is to imagine that a logarithmic operation is implicitly applied to each probability number associated with an edge. The greedy algorithm for finding a minimum spanning tree will then have to compare logarithms of two probabilities, say $\ln p_i$ and $\ln p_j$. Of course, one can compare instead p_i and p_j. Therefore the logarithmic operation (which is not even a finite process), does not explicitly have to be used.

A recent paper considered the problem of finding a minimal cost reliability ratio spanning tree [6]. Using the above approach to the problem requires comparisons of $(a \ln p_i)$ with $(b \ln p_j)$, where a and b, as well as p_i and p_j are input data of the original problem. Of course, a comparison can be executed by generating $(p_i)^a$ and $(p_j)^b$, but the length of these numbers in binary encoding is not polynomial in the input length of the original problem. Therefore, such a comparison cannot be viewed as a polynomial operation.

As shown in [6], if the finite operation of computing $(p_i)^a$ is counted as a single operation, the running time of the algorithm that finds the optimal tree is bounded by a polynomial in the number of edges of the network.

In this paper we will show that comparing $(a \ln p_i)$ with $(b \ln p_j)$ (or equivalently $(p_i)^a$ to $(p_j)^b$), can be performed in time which is polynomial in the input length of the four numbers a, b, p_i and p_j. This, in particular, yields a polynomial algorithm for the cost reliability ratio spanning tree problem of [6]. Our approach is based on bounds for linear forms in the logarithms of algebraic numbers. The interest in existence of such bounds started as early as 1923 [13], and 1935 [10]. Various papers on the subject have appeared since then, e.g. [1,2,3,4,8]. The reader is referred in particular to the excellent book by A. Baker [3], that exposes most of the results in great detail. The motivation of most of those papers has been the solution of Diophantine equations. We are not aware of any previous application to combinatorial optimization. We believe that further applications like the one discussed above and in Section 3 will follow. In this context we mention that we have employed similar bounds on the distance between algebraic numbers to obtain a polynomial algorithm for a quadratic fractional optimization problem [7].

Testing whether $a^b \geq c^d$ for integers a, b, c, d

Let $r = p/q$ be a rational number where p, q are integer, $q \neq 0$ and the g.c.d. of p and q is 1, i.e., p and q are relatively prime. The height of r is defined to be $\max(|p|, |q|)$. Throughout this paper $\ln x$ will denote the logarithm to the natural basis, while $\log x$ will denote the logarithm to the base 2. Complexity bounds of algorithms are expressed in terms of the number of elementary operations $+, -, *, /$, performed. Each such operation is assumed to consume atmost a fixed number of time units.

We will use the following result which is a specialization of [Theorem 2, 4]. (See also [Theorem 3.1, 3].)

Lemma 1. *Let $a_1, \ldots, a_k$ be positive rational numbers with heights atmost A, where $A \geq 4$, and let $b_1, \ldots, b_k$ be integers with absolute values atmost B, where $B \geq 4$. Define*

$$D = b_1 \ln a_1 + \cdots + b_k \ln a_k. \tag{1}$$

Then either $D = 0$, or

$$|D| > \Delta = B^{-c(k-1)(\ln A)^k \ln \ln A} \tag{2}$$

where $c = (16k)^{200k}$.

Motivated by the minimal cost reliability ratio spanning tree problem [6], we will focus on the special case $k = 2$. (We comment on the general case in Remark 1.) Consider the problem of determining the sign of D, (or equivalently comparing $(a_1)^{b_1}$ with $(a_2)^{-b_2}$). Let E be some rational number satisfying $|E - D| \leq \Delta/2$, where Δ is defined in (2). Then $D = 0$ if $|E| \leq \Delta/2$, $D > 0$ if $E > \Delta/2$, and $D < 0$ if $E < -\Delta/2$. Thus, the problem of determining the sign of D reduces to (efficiently) approximating D by some rational.

Let $k=2$ and let a_1, a_2, b_1, b_2 be as in Lemma 1. Define L to be the length of their input in binary encoding, i.e.,

$$L = \sum_{i=1}^{2} (\log(|b_i|+1) + \log(a_i'+1) + \log(a_i''+1)) \tag{3}$$

where $a_i = a_i'/a_i''$, and a_i', a_i'', $i=1,2$, are positive integers.

Given ε, $0<\varepsilon<1$, we will find (in time which is polynomial in L and $\log \varepsilon^{-1}$), a rational E such that $|E-D| \le \varepsilon$. Furthermore, the binary encoding of the height of E will also be polynomial in L and $\log \varepsilon^{-1}$. Note that if A and B in Lemma 1 are chosen minimal with the provision that $A \ge 4$ and $B \ge 4$, then

$$(\log A + \log B) \le L \le 2(\log(B+1) + 2(\log(A+1)). \tag{4}$$

Lemma 2. *Let p be a positive integer and let $0<\varepsilon<1$. There exists an algorithm that finds a rational q such that $|\ln p - q| < \varepsilon$ in time which is polynomial in $\log p$ and $\log \varepsilon^{-1}$. Moreover, the height of q has a binary encoding of length polynomial in $\log p$ and $\log \varepsilon^{-1}$.*

Proof. Let t be an integer such that $2^t \le p < 2^{t+1}$. If $2p \ge 2^t + 2^{t+1}$, then $3/4 \le p/2^{t+1} < 1$, and define $m=t+1$. If $2p < 2^t + 2^{t+1}$, then $1 \le p/2^t < 3/2$, and define $m=t$. Thus, we have found an integer $m \le \log p + 1$, satisfying $3/4 \le p/2^m < 3/2$.

Define $x = p/2^m - 1$. Then $-1/4 \le x < 1/2$. It takes $O(\log\log p)$ time to compute m and x.

$$\ln p = m \ln 2 + \ln(p/2^m) = m \ln 2 + \ln(1+x)$$

Therefore, it is sufficient to approximate $\ln(1+x)$ with accuracy of $\varepsilon/2$ and $\ln 2$ with accuracy of $\varepsilon/2m$. We will use the following two approximations for the logarithmic function [5].

$$\ln(1+y) = \sum_{k=1}^{n} (-1)^{k-1} \frac{y^k}{k} + R_1$$

where

$$|R_1| \le \begin{cases} \dfrac{y^{n+1}}{n+1} & \text{for } y \ge 0, \\[2ex] \dfrac{1}{1-|y|} \dfrac{|y|^{n+1}}{n+1} & \text{for } -1<y<0. \end{cases} \tag{5}$$

$$\ln y = \sum_{k=0}^{n} \frac{2}{2k+1} \left(\frac{y-1}{y+1}\right)^{2k+1} + R_2$$

where (6)

$$0 \le R_2 \le \frac{(y+1)^2}{2y(2y+3)} \left(\frac{y-1}{y+1}\right)^{2n+3} \quad \text{for } y>1.$$

To approximate $\ln(1+x)$ we use (5). We choose a minimal n such that $2^{-(n+1)}<\varepsilon/4$, i.e., let $n=\lceil \log \varepsilon^{-1}\rceil+1$. Since $-1/4\le x<1/2$, we easily verify that $|R_1|<2(2)^{-(n+1)}<\varepsilon/2$. $T_1=\sum_{k=1}^{n}(-1)^k x^k/k$ is a rational approximation of $\ln(1+x)$.

$x=p/2^m-1$, and therefore x is a rational whose height has a binary encoding of length $O(\log p)$. T_1 is a sum of $O(\log \varepsilon^{-1})$ rationals, each having height with binary encoding of $O((\log \varepsilon^{-1})(\log p))$. Therefore, the height of T_1 has an encoding of $O((\log \varepsilon^{-1})^2\,(\log p))$, and T_1 is computed in $O(\log \varepsilon^{-1})$ time.

We now turn to the approximation of $\ln 2$ in accuracy of $\varepsilon'=\varepsilon/2m$. Using formula (6) with $y=2$, we choose a minimal n such that $3^{-(2n+3)}<\varepsilon'$, i.e.,

$$n=O(\log(\varepsilon')^{-1})=O(\log \varepsilon^{-1}+\log m)=O(\log \varepsilon^{-1}+\log\log p).$$

$\ln 2$ is now approximated by

$$T_2=\sum_{k=0}^{n}\frac{2}{2k+1}3^{-(2k+1)}$$

in $O(\log \varepsilon^{-1}+\log\log p)$ time. The height of T_2 has an encoding of length

$$O((\log(\varepsilon')^{-1})^2)=O((\log \varepsilon^{-1})^2+(\log\log p)^2).$$

Finally we set $q=T_1+mT_2$. Then q is obtained in time which is of order $O(\log \varepsilon^{-1}+\log\log p)$. The length of the binary encoding of the height of q is $O((\log \varepsilon^{-1})^2(\log p))$. Furthermore,

$$\begin{aligned}|\ln p-q| &= |m(\ln 2-T_2)+(\ln(1+x)-T_1)|\\ &\le m\,|\ln 2-T_2|+|\ln(1+x)-T_1|<\varepsilon.\end{aligned}$$

The proof is now complete.

Theorem 1. *Set $k=2$ and let a_1,a_2,b_1,b_2 be as in Lemma* 1. *Let L be defined by* (3). *For each rational $0<\varepsilon<1$ there exists an algorithm that finds a rational q such that $|b_1\ln a_1+b_2\ln a_2-q|<\varepsilon$ in time which is polynomial in L and $\log \varepsilon^{-1}$. The height of q has a binary encoding of length polynomial in L and $\log \varepsilon^{-1}$.*

Proof. Using Lemma 2, approximate $\ln a_i'$ and $\ln a_i''$ by $\varepsilon_i=\varepsilon/4\,|b_i|$, $i=1,2$, to obtain q_i' and q_i'' respectively. Define $q=b_1(q_1'-q_1'')+b_2(q_2'-q_2'')$. Then,

$$\begin{aligned}|b_1\ln a_1+b_2\ln a_2-q| &= |b_1(\ln a_1'-q_1')-b_1(\ln a_1''-q_1'')\\ &\quad +b_2(\ln a_2'-q_2')-b_2(\ln a_2''-q_2'')|\\ &< 2\,|b_1|\,\varepsilon_1+2\,|b_2|\,\varepsilon_2=\varepsilon\end{aligned}$$

The result follows from Lemma 2 since $\log|b_i|$, $\log a_i'$ and $\log a_i''$, are bounded above by L, and $\log \varepsilon_i^{-1}=O(\log \varepsilon^{-1}+L)$, $i=1,2$.

Theorem 2. *Let a_i', a_i'', $i=1,2$, be positive integers, and define $a_i=a_i'/a_i''$. Let b_1*

and b_2 *be integers and define*

$$D = b_1 \ln a_1 + b_2 \ln a_2$$

The sign of D *can be determined in time which is polynomial in the input length,* L*, given by* (3).

Proof. Using Lemma 1 and (4), we note that if $D \neq 0$, then

$$|D| > \Delta_1 = 2^{-c'L^4}$$

where c' is some constant. Choose $\varepsilon = \Delta_1/2$ and apply the algorithm suggested by Theorem 1, to obtain a rational q such that $|D-q| < \varepsilon$. We have already observed that $D=0$ if $|q| \leq \Delta_1/2$, $D>0$ if $q > \Delta_1/2$, and $D<0$ if $q < -\Delta_1/2$. The time needed to find q is polynomial in L and $\log \varepsilon^{-1} = O(L^4)$. Thus, the proof is complete.

As a consequence of the above theorem we have the following.

Corollary 1. *Given integers* a, b, c, d, *there exists a polynomial algorithm to determine whether* $a^b \geq c^d$.

Remark 1. Although our concern in this paper has been the testing of the query "is $a^b \geq c^d$?", (i.e., the case $k=2$ in Lemma 1), we note that our approach can easily be extended to the general case to determine the sign of $b_1 \ln a_1 + \cdots + b_k \ln a_k$ (a_i, b_i, $i = 1, \ldots, k$, integer), in polynomial time in terms of the input length, provided k is fixed. Consequently, given integers $a_1, \ldots, a_k, b_1, \ldots, b_k, c_1, \ldots, c_k, d_1, \ldots, d_k$, there exists a polynomial algorithm to determine whether $a_1^{b_1} \cdots a_k^{b_k} \geq c_1^{d_1} \cdots c_k^{d_k}$ for any *fixed* value of k.

To our knowledge the existence of a test which is also polynomial in k is an open problem.

Remark 2. We note that the constant coefficient c in Lemma 1 takes on the value $(32)^{400}$ for $k=2$. A lower bound on D (for $k=2$) with a smaller constant term c is given in [Proposition 6, 12].

The minimal cost reliability ratio spanning tree problem

Let $G = (V, E)$ be an undirected graph with V and E as its sets of vertices and edges respectively. For each edge $i \in E$ let p_i ($0 < p_i \leq 1$) and c_i, respectively, be the probability of functioning and the nonnegative integral cost. Let Ω be the set of all spanning trees in G. For each $T \in \Omega$ define $C(T) = \sum_{i \in T} c_i$, $R(T) = \prod_{i \in T} p_i$, and $S(T) = \sum_{i \in T} -\ln p_i$. The minimal cost reliability ratio spanning tree problem, considered in [6] is:

$$\operatorname*{Min}_{T\in\Omega} m(T) = \frac{C(T)}{R(T)}. \tag{7}$$

Define the bicriteria spanning tree problem

$$\operatorname*{Min}_{T\in\Omega} (S(T), C(T)). \tag{8}$$

It is shown in [6] that a spanning tree minimizing (7) is in the set H, defined as the set of trees corresponding to extreme points of the convex hull of the bicriteria space (8). A systematic search that produces the set H is given in [6]. A step of this search requires finding a minimal spanning tree with edge weights

$$w_i = p_i^{(C(T_l)-C(T_k))} \left| \frac{R(T_k)}{R(T_l)} \right|^{c_i} \tag{9}$$

where T_k and T_l are given trees in Ω.

Counting the operation $p_i^{c_j}$ as a single operation (as done in [6]), the above minimal spanning tree can be found in $O(|E| \log\log |V|)$ time [15]. Since the binary encoding of $p_i^{c_j}$ is not polynomial in the input length, the entire process cannot be viewed polynomial.

The results of the previous section enable us to overcome this difficulty and compare weights w_i and w_j as in (9) without explicitly performing the operation $p_i^{c_j}$. This will yield a polynomial procedure for (7). Given weights w_i and w_j we note that $w_i \ge w_j$ if and only if

$$(C(T_l)-C(T_k))\ln(p_i/p_j) + (c_i - c_j)\ln(R(T_k)/R(T_l)) \ge 0. \tag{10}$$

By Theorem 2 we can determine whether $w_i \ge w_j$ in polynomial time in the length of the problem input. Therefore, the greedy algorithm with weights w_i, $i \in E$, can be applied in polynomial time.

As a final comment we note in passing that the recent work of Gusfield [11] implies that the frontier H contains $O(|E|^{3/2})$ trees. Therefore, H can be obtained in $O(|E|^{5/2} \log\log |V|)$ applications of the greedy algorithm.

References

[1] A. Baker, Linear forms in the logarithms of algebraic numbers I, II, III, IV, Mathematika 13 (1966) 204–216; 14 (1967) 102–107, 220–228; 15 (1968) 204–216.

[2] A. Baker, A sharpening of the bounds for linear forms in logarithms I, II, III, Acta Arithmetica 21 (1972) 117–129; 24 (1973) 33–36; 27 (1974) 247–252.

[3] A. Baker, Transcendental Number Theory (Cambridge Univ. Press, Cambridge, 1975).

[4] A. Baker, The theory of linear forms in logarithms, in: A. Baker and D. Masser, eds., Transcendence Theory: Advances and Applications (Academic Press, London and New York, 1977).

[5] L. Bers, Calculus (Holt, Rinehart & Winston, New York, 1969).

[6] R. Chandrasekaran, Y.P. Aneja and K.P.K. Nair, Minimal cost reliability ratio spanning tree, Annals Discrete Math. 11 (1981) 53–60.

[7] R. Chandrasekaran and A. Tamir, Optimization problems with algebraic solutions: quadratic fractional programs and ratio games, Math. Programming, to appear.

[8] N.I. Feldman, Estimation of a linear form in the logarithms of algebraic numbers, Mat. Sbornik 77 (1968) 423-426. English translation in: Math. USSR Sbornik 6 (1968) 393-406.

[9] H. Frank and I.T. Frisch, Communication, Transmission and Transportation Networks (Addison-Wesley, Reading, MA, 1971).

[10] A.O. Gelfond, On the approximation of transcendental numbers by algebraic numbers, Doklady Akad. Nauk SSSR 2 (1935) 177-182.

[11] D.M. Gusfield, Sensitivity analysis for combinatorial optimization, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, May 1980 (Memo. No. UCB/ERL M80/22).

[12] M. Mignotte and M. Waldschmidt, Linear forms in two logarithms and Schneider's Method, Math. Ann. 231 (1978) 241-267.

[13] D. Morduchai-Boltovskoj, Sur le logarithme d'un nombre algebrique, C.R. Acad. Sci. (Paris) 176 (1923) 724-727.

[14] R.C. Prim, Shortest connected networks and some generalizations, Bell Syst. Tech. J. 36 (1957) 1389-1401.

[15] A.C. Yao, An $O(|E| \log\log |V|)$ algorithm for finding minimum spanning tree, Inform. Process. Lett. 4 (1975) 21-23.