

DECIPHERING THRESHOLD FUNCTIONS OF k -VALUED LOGIC[†])

N. Y. Zolotykh and V. N. Shevchenko

The problem of deciphering threshold functions of k -valued logic of n arguments is considered. A polynomial deciphering algorithm is proposed which, given n , uses at most $O(\log^n(k+1))$ appeals to the oracle.

Introduction

Consider the following two-person game. On the playing board $B(2, k)$ consisting of all points $(x, y) \in Z^2$ such that $0 \leq x \leq k-1$, $0 \leq y \leq k-1$, the first player chooses two distinct points and finds the equation $ax + by = c$ of the line through them. The second player must find the equation of this line by choosing points $(x_i, y_i) \in B(2, k)$ and asking the first player whether it is true that $ax_i + by_i \leq c$. From the point of view of the second player, it is natural to find a strategy guaranteeing the answer with a small number of questions and an admissible time for computing the coordinates (x_i, y_i) . Generalizing this situation, we introduce the following notation and notions [1, 2].

Let k and n be positive integers, $B(n, k)$ be the set of integral vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)$ such that $0 \leq x_j \leq k-1$ ($j = 1, 2, \dots, n$), $f(\mathbf{x})$ be a function mapping $B(n, k)$ into the set $\{0, 1\}$, $M_0(f)$ and $M_1(f)$ be the sets of vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)$ such that $f(\mathbf{x})$ equals 0 and 1, respectively, i.e., $M_0(f) = \{\mathbf{x} \in B(n, k) \mid f(\mathbf{x}) = 0\}$, $M_1(f) = \{\mathbf{x} \in B(n, k) \mid f(\mathbf{x}) = 1\}$, and $N_i(f)$ be the set of the extreme points of the convex hull of $M_i(f)$ ($i = 0, 1$).

A function $f(\mathbf{x})$ is a *threshold function* if there exist real numbers a_i ($i = 0, 1, \dots, n$) such that

$$M_0(f) = \left\{ \mathbf{x} \in B(n, k) \mid \sum_{i=1}^n a_i x_i \leq a_0 \right\} \quad (1)$$

The set of all threshold functions defined on $B(n, k)$ is denoted by $F(n, k)$.

[†]) This research was supported by the Russian Foundation for Basic Research (Grant 94-01-00491).

Suppose that, to a function $f \in F(n, k)$, there corresponds some oracle that allows us to compute $f(\mathbf{x})$ given $\mathbf{x} \in B(n, k)$. We call such oracle the M -oracle. We understand the M -deciphering of the function $f \in F(n, k)$ to be a procedure for finding, with the help of the M -oracle, the numbers a_0, a_1, \dots, a_n for which (1) is satisfied.

An algorithm A for M -deciphering threshold functions is said to be *polynomial* if, for each function $f \in F(n, k)$, both the number $\tau(A)$ of appeals to the oracle and the number $\rho(A)$ of required operations are bounded above by some polynomials in n and $\log(k+1)$. (Throughout this paper, \log denotes the base 2 logarithm.) In [2], it is proved that no polynomial algorithm exists for M -deciphering threshold functions. With a fixed dimension n , a polynomial algorithm is called *quasipolynomial*.

In [1], it is observed that to define a threshold function f , it suffices to define either the set $N_0(f)$ or the set $N_1(f)$, and it is shown that

$$|N_i(f)| \leq (2 \log(k+1))^n \quad (i = 0, 1).$$

In [3], a quasipolynomial algorithm is proposed for the construction of the convex hull of the set M of integral solutions to a system of linear inequalities based on the Lenstra algorithm [4] for finding a point $\mathbf{x} \in M$ and on the bound in [5] for the number of the extreme points and faces of the convex hull of M . These results are essentially used in the quasipolynomial algorithm A_1 for deciphering threshold functions which was developed in [2]. Later [6, 7], the upper bound of $\tau(A_1)$ is lowered on using refinement [3, 8, 9] of upper bounds on $|N|$; in [7], it has the form

$$\tau(A_1) = O((\log(k+1))^{n+\lfloor n/2 \rfloor (n-1)}).$$

Note that no further decrease in the exponent seems possible with this approach since the bound on the number of extreme points obtained in [10] is best possible in order [11, 12].

We propose a quasipolynomial algorithm A_2 for deciphering $f \in F(n, k)$ such that $\tau(A_2) = O(\log^n(k+1))$.

To construct the algorithm, we use the results of [13], where some more informative oracle is associated with a threshold function. We call this oracle the E -oracle. It is defined as follows. Given any integers a_0, a_1, \dots, a_n , the E -oracle of the function $f \in F(n, k)$ answers "yes" if $M_0(f)$ satisfies (1); otherwise, it answers "no" and produces a point $z = (z_1, z_2, \dots, z_n)$ in $B(n, k)$ such that either $\sum_{i=1}^n a_i z_i > a_0$ and $f(z) = 0$ (positive counterexample) or $\sum_{i=1}^n a_i z_i \leq a_0$ and

$f(z) = 1$ (negative counterexample). We understand the E -deciphering of a function f in the class $F(n, k)$ to be a sequence of appeals to the E -oracle of f which results in the answer “yes.”

When constructing the algorithm A_2 in Section 2, we show that rather than appealing once to the E -oracle, it suffices to appeal to the M -oracle $O(\log^{n-1}(k+1))$ times.

1. Preliminary Results

We first reformulate the following result in [10].

Lemma 1. *Let P be a polyhedron defined by a system of m linear inequalities with integral coefficients whose absolute values do not exceed α , let $M(P)$ be the intersection of P with the integer lattice, and let $N(P)$ be the set of the extreme points of the convex hull of $M(P)$. Then, for any fixed m and n , it is true that $|N(P)| = O(\log \alpha^{n-1})$.*

Lemma 2. *Suppose the coefficients a_0, a_1, \dots, a_n with the length of the binary notation bounded above by a polynomial in $\log(k+1)$ are presented to the E -oracle. Then, in time polynomial in $\log(k+1)$, each such appeal to the E -oracle can be reduced to $O(\log^{n-1}(k+1))$ questions to the M -oracle (n is fixed).*

PROOF. First, using the algorithm in [3], find the set $N(a_0, a_1, \dots, a_n)$ of vertices of the convex hull of the set

$$\left\{ \mathbf{x} \in B(n, k) \mid \sum_{i=1}^n a_i x_i \leq a_0 \right\}.$$

Next, with the help of the M -oracle, consecutively checking the value of f at each of these points, find a point $z \in N(a_0, a_1, \dots, a_n)$ such that $f(z) = 1$ or find the absence of such points. Obviously, in the first case, z is a negative counterexample. In the second case, construct the set $N'(a_0, a_1, \dots, a_n)$ of vertices of the convex hull of the set

$$\left\{ \mathbf{x} \in B(n, k) \mid \sum_{i=1}^n a_i x_i > a_0 \right\}$$

and appeal to the M -oracle at each point in $N'(a_0, a_1, \dots, a_n)$ unless a point $z' \in N'(a_0, a_1, \dots, a_n)$ is found such that $f(z') = 0$ or find the absence of such

points. In the first case, z' is clearly a positive counterexample whereas, in the second case, as follows from [1],

$$M_0(f) = \left\{ \mathbf{x} \in B(n, k) \mid \sum_{i=1}^n a_i x_i \leq a_0 \right\}$$

i.e., the function f is deciphered.

Thus, we have reduced one appeal to the E -oracle to a series of appeals to the M -oracle. We give an upper bound for the number of such appeals and the number of required arithmetic operations. In [1], it is shown that, given a fixed n , for each $f \in F(n, k)$, there exist integers $b_j (j = 0, 1, \dots, n)$ such that $|b_j|$ is bounded by a polynomial in k and

$$M_0(f) = \left\{ \mathbf{x} \in B(n, k) \mid \sum_{i=1}^n b_i x_i \leq b_0 \right\}$$

It follows from this and Lemma 1 that, for given a fixed n ,

$$|N(a_0, a_1, \dots, a_n)| = |N(b_0, b_1, \dots, b_n)| = O(\log^{n-1}(k+1)).$$

Similarly, it can be shown that $|N'(a_0, a_1, \dots, a_n)| = O(\log^{n-1}(k+1))$. Therefore, the number of appeals to the M -oracle required for reducing one appeal to the E -oracle does not exceed $O(\log^{n-1}(k+1))$.

Now we give an upper bound for the number of arithmetic operations executed. To find points in the set $N(b_0, b_1, \dots, b_n)$, we made use of the algorithm proposed in [3].

Given n , the running time of this algorithm is bounded by a polynomial in $\log(\alpha + 1)$, where $\alpha = \max\{k, a_0, a_1, \dots, a_n\}$. Since, for given n , the length of the binary notation of the coefficients a_0, a_1, \dots, a_n is bounded above by a polynomial in $\log(k+1)$, the number of operations required for reducing one appeal to the E -oracle to a series of appeals to the M -oracle is bounded above by a polynomial in $\log(k+1)$. Lemma 2 is proved.

2. Main Results

We provide some clarifications to the algorithm A_3 for E -deciphering threshold functions which was proposed in [13].

The algorithm A_3 uses an algorithm of finding a point in a convex body $W \subseteq \mathbb{R}^n$ defined by the *separated oracle* (see, for example, [14]). The separated oracle for W , for given a point $b \in \mathbb{R}^n$, answers “yes” if $b \in W$; otherwise, the oracle answers “no” and produces the coefficients of a hyperplane separating b from W .

Without loss of generality (see [13]), we may assume that $a_0 = 1$ in (1). Hence, to each threshold function f , there corresponds a convex body $W(f) = \{u \in \mathbb{R}^n \mid M_0(f) = \{\mathbf{x} \mid (u, \mathbf{x}) \leq 1\}\}$ in the space \mathbb{R}^n .

Thus, the problem of E -deciphering a threshold function can be stated as a problem of finding a point in $W(f)$.

In [13], a way to construct the separated oracle for $W(f)$ based on the E -oracle for f is indicated. The ellipsoid algorithm [14, 15] can be used as an algorithm for finding a point in a convex body.

We state the result of [13] as a theorem.

Theorem 1 [13]. *There exists an algorithm A_3 for E -deciphering threshold functions such that, for given a fixed n , $\tau(A_3) = O(\log k)$ and both $\rho(A_3)$ and the length of the binary notation of each coefficient a_0, a_1, \dots, a_n under all the appeals to the E -oracle are bounded above by some polynomials in $\log(k + 1)$.*

Theorem 1 and Lemma 2 imply the following

Theorem 2. *Given a fixed n , there exists a polynomial algorithm for M -deciphering threshold functions of k -valued logic of n arguments which uses at most $C_n \log^n(k + 1)$ questions on the value of the function $f(\mathbf{x})$ at a point \mathbf{x} , where C_n is a constant depending only on n .*

References

1. V. N. Shevchenko (1985) On some functions of many-valued logic connected with integer programming (in Russian), *Metody Diskret. Anal.* **42**, 99–108.
2. V. N. Shevchenko (1987) On deciphering of threshold functions of many-valued logic (in Russian), in: *Combinatorial-Algebraic Methods and Their Applications*, Gor'kov. University, Gor'kiĭ, pp. 155–167.
3. V. N. Shevchenko (1985) The algebraic approach in integer programming (in Russian), *Kibernetika* (Kiev), No. 4, 36–41.
4. H. W. Lenstra, Jr. (1983) Integer programming with a fixed number of variables, *Math. Oper. Res.* **8**, No. 4, 538–548.
5. V. N. Shevchenko (1985) Convex polyhedral cones, comparison systems, and valid cuts in integer programming (in Russian), in: *Combinatorial-Algebraic*

- Methods in Applied Mathematics*, Gor'kov. University, Gor'kiĭ, pp. 109–119.
6. V. N. Shevchenko and S. I. Veselov (1989) Deciphering functions of many-valued logic (in Russian), *Theory and Program Realization of Discrete Optimization Methods*, Institute of Cybernetics, Kiev, pp. 30–34.
 7. T. Hegedus (1994) Geometrical concept of learning and convex polytopes, *Proceedings of the 7th Annual Conference on Computational Learning Theory*, ACM Press, New Brunswick, New York.
 8. V. N. Shevchenko (1985) Upper bounds for the number of extreme points in integer programming (in Russian), *Mathematical Problems in Cybernetics*. Vol. 4, Nauka, Moscow, pp. 65–72.
 9. A. Yu. Chirkov and V. N. Shevchenko (1993) On the number of vertices in the convex hull of the intersection of a polyhedron with the integer lattice (in Russian), VINITI, No. 2165-B93, Moscow.
 10. Cook W., Hartmann M., Kannan R., and McDiarmid C. (1992) On integer points in polyhedra, *Combinatorica* **12**, No. 1, 27–37.
 11. S. I. Veselov (1984) A lower bound for the number of irreducible and boundary points for two problems of discrete programming (in Russian), VINITI, No. 619-B84, Moscow.
 12. A. Yu. Chirkov (1994) A lower bound for the number of vertices in the convex hull of the intersection of a polyhedron with the integer lattice (in Russian), VINITI, No. 1361-B94, Moscow.
 13. W. Maas and G. Turán (1991) How fast can a threshold gate learn? *IIG-Report Ser. Rep. 321*, Graz University of Technology.
 14. A. Schrijver (1986) *Theory of Linear and Integer Programming*, John Wiley and Sons, Chichester.
 15. L. G. Khachiyan (1979) A polynomial algorithm in linear programming (in Russian), *Dokl. Akad. Nauk SSSR* **244**, No. 5, 1093–1096.

Nizhniĭ Novgorod State University
Gagarin pr., 23
Nizhniĭ Novgorod 603600
RUSSIA

TRANSLATED BY S. I. SUSLOV