ГОДИШНИК НА СОФИЙСКИЯ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
Книга 1 — Математика
Том 84, 1990

ANNUAIRE DE L'UNIVERSITE DE SOFIA „ST. KLIMENT OHRIDSKI“

FACULTE DE MATHEMATIQUES ET INFORMATIQUE
Livre 1 — Mathématiques
Tome 84, 1990

ON THE CALCULATION OF LINEAR PROGRAMMING BOUND FOR ERROR CORRECTING CODES

KRASSIMIR MANEV, MAYA PETKOVA, RADA GOLEMANOVA

Красимир Манев, Майа Петкова, Рада Големанова. О ВЫЧИСЛЕНИИ ГРАНИЦЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ДЛЯ КОДОВ, ИСПРАВЛЯЮЩИХ ОШИБОК

Граница линейного программирования для кодов, исправляющих ошибок, не является формулой, а результатом вычислительной процедуры, которой нелегко выполнить даже при помощи компьютера. Настоящая работа посвещена разработанной авторами программе для точного вычисления этой границы при заданными параметрами кода. Программа состоит из двух независимых частей, которые можно применять и самостоятельно. Первая часть вычисляет значения полиномов Кравчука, при помощи которых составляется задача линейного программирования. Вторая реализует рациональный симплекс-метод для решении этой задачи. Обе части применяют рациональную арифметику с предварительно заданной точностью.

Krassimir Manev, Maya Petkova, Rada Golemanova. ON THE CALCULATION OF LINEAR PROGRAMMING BOUND FOR ERROR CORRECTING CODES

The bound of linear programming for error correcting codes is not a formula but a result of a computational procedure difficult to perform even on powerful computers. In this paper a program for calculating the bound for given parameters of the code is described. The program comprises two independent parts, which can be used separately also. The first of them calculates the values of the Krawtchouk polynomials for given parameters and builds the linear programming problem. The other implements a rational simplex-method for solving that problem. Both of them use rational arithmetic with preliminary defined precision.

Let's remind some definitions and results from [1].

Definition. An (n, M, d) code is a set of M vectors of length n (with components from some field F), such that any two vectors differ in at least d places and

d is the smallest number with this property.

Definition. If L is an (n, M, d) code, the distance distribution of L consists of the numbers $B_0, B_1, \ldots, B_n$, where

$$B_i = \frac{1}{M}\{\text{number of ordered pairs of codewords } u, v, \text{ such that } \operatorname{dist}(u, v) = i\}.$$

With $\operatorname{dist}(u, v)$, called distance between u and v, we denote the number of places, where vectors u and v differ. Note that $B_0 = 1$ and $B_0 + B_1 + \cdots + B_n = M$.

Definition. For any positive integer n the Krawtchouk polynomial of the real variable x $P_k(x; n) = P_k(x)$ is defined by

$$P_k(x; n) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j}, \quad k = 0, 1, 2, \ldots,$$

where the binomial coefficients are defined as follows:

$$\binom{x}{m} = \begin{cases} \dfrac{x(x-1)\ldots(x-m+1)}{m!}, & \text{if } m \text{ is a positive integer} \\ 1, & \text{if } m = 0 \\ 0, & \text{otherwise,} \end{cases}$$

where $m! = 1.2.3\ldots(m-1).m$, $0! = 1$. Thus $P_k(x; n)$ is a polynomial in x of degree k. If there is no danger of confusion, we omit the n.

Definition. For any $(n+1)$-tuple $\{B_0, B_1, \ldots, B_n\}$ with $\sum_{i=0}^{n} B_i \neq 0$ we call $\{B'_0, B'_1, \ldots, B'_n\}$ the McWilliams transformation of the B_i, $i = 0, 1 \ldots, n$, where

$$B'_k = \frac{1}{M} \sum_{i=0}^{n} B_i P_k(i), \quad k = 0, 1, \ldots, n,$$

$P_k(x)$ is a Krawtchouk polynomial and M is the number of codewords (M also is called a size of the code).

Probably, the most basic problem in coding theory is to find the largest code of a given length n and a minimum distance between codewords d.

Sometimes it is possible to use linear programming to obtain excellent bounds on the size of a code with a given distance distribution.

Let L be an (n, M, d) code, in which the distances between the codewords are $\tau_0 = 0 < \tau_1 < \ldots < \tau_s$. Let $\{B_i\}$ be the distance distribution of L. Thus $B_0 = 1$, $B_{\tau_j} > 0$ $(j = 1, \ldots, s)$ and $B_i = 0$ otherwise. Also $M = 1 + \sum_{j=1}^{s} B_{\tau_j}$. Therefore the transformed distribution $\{B'_k\}$ is given by

$$B'_k = \frac{1}{M} \sum_{j=0}^{s} B_{\tau_j} P_k(\tau_j), \quad k = 0, \ldots, n,$$

where $P_k(x)$ is a Krawtchouk polynomial. Also $B'_0 = 1$, $\sum_{k=0}^{n} B'_k = 2^n/M$ and $B'_k \geqq 0$ for $k = 0, \ldots, n$.

Thus if L is any code with distances τ_j, $j = 0, \ldots, s$, between the codewords, then $B_{\tau_1}, \ldots, B_{\tau_s}$ is a feasible solution to the following linear programming problem:

(1) Choose $B_{\tau_1}, \ldots, B_{\tau_s}$, so as to maximize $\sum_{j=1}^{s} B_{\tau_j}$, subject to

$$B_{\tau_j} \geqq 0, \quad j = 1, \ldots, s,$$

$$\sum_{j=1}^{s} B_{\tau_j} P_k(\tau_j) \geqq -\binom{n}{k}, \quad k = 1, \ldots, n.$$

Therefore we have the following theorem:

Theorem [2]. *If $B^*_{\tau_1}, \ldots, B^*_{\tau_s}$ is an optimal solution to* (1), *then* $1 + \sum_{i=1}^{s} B^*_{\tau_j}$ *is an upper bound on the size of L.*

Note that (1) certainly has a feasible solution: $B_{\tau_j} = 0$ for all i.

The using of the above mentioned result for practical purposes is very difficult because of the following reasons:

a) the Theorem does not give an explicit formula for the upper bound. The finding of a particular value of this bound requires to work out and resolve nontrivial problem of the linear programming (LPP);

b) both the working out and the solving of the LPP require calculations which exclude not only a manual treating but an ordinary computer program too. Indeed, the values of the Krawtchouk's polynomials grow up so rapidly that it is impossible to calculate them (even for not very large parameters) on any computer without using special arithmetic programs. Moreover, having these values calculated, new difficulties arise when we solve the LPP with such enormous coefficients.

That is why we have developed and implemented a program, which will find the particular value of the linear programming bound for given parameters.

The main ideas, followed by us, have been:

1) To work out the matrix of the LPP using the programs from the package COMPACK [3], which performs arithmetic operations with arbitrary big integers.

2) To extent the package with "big rational" arithmetic programs and to implement a "rational" simplex method [4], which allows to find the upper bound without loosing precision.

Naturally, the proposed program `lpb` (linear programming bound) consists of two essential parts. The first part, called `kravtchuk`, input the parameters n and d, where n is the length of the code and d — the required minimal distance. As a result this part constructs a matrix of the LPP. The second part, called `rsimplex`, implements the classical simplex method, using "big rational" arithmetic and gives the exact optimal solution of the LPP.

The interface between these two parts can be modified through adding or/and deleting some conditions, arising from other combinatorial or algebraic reasons.

Each of the two parts of the program `lpb` is implemented as a separate program too. So they could be used for solving other problems — **`kravtchuk`** for different calculations connected with Krawtchouk's polynomials and **`rsimplex`** for precise solving of rational linear programming problems and as a preliminary step of more complex programs for solving problems of the integer linear programming.

Now the package COMPACK contains the following new standard programs, which could be useful:

- the programs for "big rational" arithmetic — **`rat_add`**, **`rat_subtr`**, **`rat_mult`**, **`rat_div`** and the program for comparing two "big rational" numbers — **`rat_testx`**;
- the programs connected with the equivalent transformations of the "big rational" numbers — **`modul`**, **`fact`**, **`factxy`**, **`twotry`**, **`twofactor`**, etc.;
- the program **`diag`** — for diagonalisation of a matrix with "big rational" elements.

The programs are written in programming language C and are implemented on IBM PC under MS DOS. They are portable in all UNIX–like operating systems and systems with standard C compiler. The maximal size of a "big rational" number is defined as a preprocessor parameter and could be changed before compilation, if necessary.

As an example let consider the parameters $n = 19$, $d = 8$. In this case the program gives (with some modifications) the following LPP:

Choose $B_8, \ldots, B_{19}$, *so as to maximize* $\sum_{j=8}^{19} B_j$, *subject to* $M.(B_8, \ldots, B_{19}) > A$, where M is the matrix

$$\begin{vmatrix}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
3 & 1 & -1 & -3 & -5 & -7 & -9 & -11 & -13 & -15 & -17 & -19 \\
-5 & -9 & -9 & -5 & 3 & 15 & 31 & 51 & 75 & 103 & 135 & 171 \\
-23 & -9 & 9 & 23 & 25 & 7 & -39 & -121 & -247 & -425 & -663 & -969 \\
4 & 36 & 36 & 4 & -44 & -76 & -44 & 116 & 484 & 1156 & 2244 & 3876 \\
76 & 36 & -36 & -76 & -36 & 84 & 204 & 132 & -468 & -2108 & -5508 & -11628 \\
28 & -84 & -84 & 28 & 140 & 92 & -196 & -532 & -196 & 2380 & 9996 & 27132 \\
-140 & -84 & 84 & 140 & -28 & -260 & -156 & 572 & 1300 & -884 & -13260 & -50388 \\
-98 & 126 & 126 & -98 & -210 & 78 & 494 & 78 & -1794 & -2210 & 11934 & 75582 \\
154 & 126 & -126 & -154 & 154 & 286 & -286 & -858 & 858 & 4862 & -4862 & -92378
\end{vmatrix}$$

and

$$A = (-136, -20, -172, -970, -3877, -11629, -27133, -50389, -75583, -92379).$$

Using 64-bits arithmetic on 80286, the solution

$$(659/8, 0, 0, 85/4, 93/8, 83/4, 0, 0, 0, 0, 0, 0)$$

has been calculated by the program in 50 minutes. The value of the linear programming bound in this case is 136.

REFERENCES

1. McWilliams, F. J., N. J. A. Sloane. The Theory of Error Corecting Codes. North Holland, Amsterdam, 1977.
2. Delsart, P. Bounds for Unrestricted Codes by Linear Programming. — Philips Res. Reports, **27**, 1972, 272–289.
3. Manev, K. N. A Portable Package of Standard Combinatorial Programs. — Сб. докладов Научно–техн. семинара „Проблемы мобильности инструментально–технолог. средств программирования", КНП-8, Варна, 1987, 157–162.
4. Dantzig, G. B., A. Ordeu, P. Wolf. The Generalized Simplex Method for Minimizing a Linear Form under Linear Inequality Constraints. — Pacific J. of Math., **5**, No 2, 1955, 1983–1995.

Received 30.03.1991