Fast Unimodular Counting

John Mount*
jmount@mzlabs.com

March 4, 1998

Abstract

This paper describes methods for counting the number of non-negative integer solutions of the system $Ax = b$ when A is a non-negative totally unimodular matrix and b an integral vector of fixed dimension. The complexity (under a unit cost arithmetic model) is strong in the sense that it depends only on the dimensions of A and *not* on the size of the entries of b. For the special case of "contingency tables" the run time is $2^{O(\sqrt{d}\log d)}$ (d the dimension of the polytope). The method is complementary to Barvinok's in that our algorithm is effective on problems of high dimension with a fixed number of (non-sign) constraints whereas Barvinok's algorithms are effective on problems of low dimension and an arbitrary number of constraints.

1 Introduction

In this paper we are concerned with determining the number of non-negative integer solutions, x, to the linear system $Ax = b$ where A is a non-negative totally unimodular matrix. Even the restricted "contingency table" version of this problem is $\sharp P$ hard in general.[7] The method applies even when the dimension is not fixed (only the number of equality constraints need to be fixed). In the application of interest the number of equality constraints and dimension are simultaneously fixed.

It was shown by Barvinok [2, 1] that counting the number of lattice points in an *integral* polytope is solvable in polynomial time when the dimension is fixed. This method is somewhat difficult to implement (requiring the solution of fixed dimension integer programs and extensive use of triangulations) so there is still some room for specialized techniques for specific problems. We remark that the 4×4 transportation (or magic square) polytope is not simple and has a set of vertices that form a non-primitive simplex. Since Barvinok's algorithm is based on the additivity of valuations on *simple* cones, this means Barvinok's algorithm must triangulate and may encounter non-primitive cones.

We also would like to point out the fascinating literature on the dependencies of counting functions (algebraic relations obeyed by the number of non-negative integer

*Thanks: Mathematical Sciences Research Institute DMS9022140

solutions to various linear systems- allowing the computation of solutions for new systems with no additional counting).[4, 17, 18] While there is a theory how to solve counting problems directly using algebraic information ("Todd classes" [8]), the less delicate method of using the algebraic dependencies directly on counts (and using brute force to compute the required values) seems fairly effective [13].

Throughout the paper, A is a non-negative totally unimodular nonnegative m row by n column matrix of full row rank. Without loss of generality we will assume that there is no nonzero nonnegative vector x such that $Ax \leq 0$. We are interested in computing the number of non-negative integral solutions to $Ax = b$, $b \in \mathbb{Z}^m \geq 0$. Set $B = \max_i b_i$, and let $\sharp(\{...\})$ denote the number of items in the set $\{...\}$.

2 Method

We are striving for a "strong" solution to the counting problem. That is: an algorithm with run time, under the unit cost arithmetic model on numbers of size commensurate with the desired answer, depends only on the order of A and not on the magnitude of the entries of b.

We show that for a fixed matrix A the function $f_A(b) : \mathbb{Z}^m \to \mathbb{Z}$ that computes the number of nonnegative integral solutions to the system $Ax = b$ is a piecewise polynomial in b with degree $n - m$ and no more than $\left(m\binom{n}{m}\right)^m$ pieces. Our method is: for a given b determine the piece and derive the polynomial p on that section. We then evaluate $p(b)$ to get the desired value of $f_A(b)$. Determining which polynomial to use is simple: our decomposition is into polyhedral cones, each defined by no more than $m\binom{n}{m}$ linear inequalities. To find the cone our problem falls in we do not need to compute the entire decomposition (or "fan"), but only inspect the $m\binom{n}{m}$ linear inequalities. The required polynomial is derived by solving $\binom{n}{m}$ counting problems involving numbers of size no more than $m^{m/2}$ and then applying an interpolation formula.

To solve the $\binom{n}{m}$ smaller problems required by the interpolation formula we exhibit two methods that have proved useful in practice. The first method is a divide and conquer approach that has proven useful for problems where b has small norm. The other method we call the "even/odd" approach and can solve the needed sub-problems in time $2^{O(m \log m)}$ in general. For the "magic square" case which has dimension $d = O(m^2)$ this yields a runtime of $2^{O(\sqrt{d} \log d)} \log B$ using the "even/odd" approach directly and a runtime of $2^{O(\sqrt{d} \log d)}$ using the interpolation approach outlined above (even after allowing for the time to identify which cone we are in and perform the interpolation).

2.1 Piecewise Polynomiality

Throughout this note, P will be a lattice polytope, that is a polytope such that all vertices have integer coordinates. A famous theorem of Ehrhart is that as k ranges over the non-negative integers the number of integral points in the polytope kP is a polynomial in k with degree the dimension of the smallest affine space containing P (see [11]).

In fact the result immediately strengthens into an analogue of the so called "mixed volumes" of convex geometry (the symbol $+$ denotes the Minkowski sum).

Theorem 1 (McMullen,Bernštein) *For any lattice polytopes $P_1, \cdots, P_M \subset \mathbb{R}^N$ and integers $k_1, \cdots, k_M \ \geq 0$*

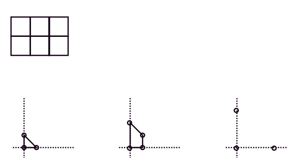
$$\sharp(k_1 P_1 + \cdots + k_M P_M) =$$

$$\sum_{r_1+\cdots+r_M \leq N} C(P_1, r_1, \cdots, P_M, r_M) k_1^{r_1} \cdots k_M^{r_M}$$

where the $C(P_1, r_1, \cdots, P_M, r_M)$ are constants depending only on $P_1, r_1, \cdots, P_M, r_M$.

The counting function $f_A(b)$ (mentioned above) is not itself a single polynomial for the simple reason that there are A, b_1, b_2 such that the polytopes $Ax = b_1,\ x \geq 0$ and $Ax = b_2,\ x \geq 0$ do not Minkowski sum to the polytope $Ax = b_1 + b_2,\ x \geq 0$. An example is the set of contingency tables given by

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}, b = \begin{bmatrix} r_1 \\ r_2 \\ c_1 \\ c_2 \end{bmatrix}:$$



For $\sigma \subset \{1, 2, \cdots, n\}$ of cardinality m, let A_σ be the $m \times m$ matrix formed by the columns of A indexed by σ. Similarly for any vector v let v_σ be the entries of v indexed by σ. Let Γ be the set of σ such that $\det(A_\sigma) \neq 0$. Let Ψ be the set of all $r \in \mathbb{R}^m$ such that there exists $\sigma \in \Gamma$ such that r is a row of A_σ^{-1}. This divides the positive orthant in

$\mathbb{R}^m$ into a collection of equivalence classes such that for $b_1, b_2 \in \mathbb{R}^m$ we have $b_1 \equiv b_2$ iff $r \cdot b_1 \geq 0 \;\leftrightarrow r \cdot b_2 \geq 0 \;\; \forall r \in \Psi$. We are interested in the largest dimensional cones in this "fan." We say b_1, b_2 are in the same cone iff $(r \cdot b_1) \times (r \cdot b_2) \geq 0 \;\; \forall r \in \Psi$.

Theorem 2 *If b_1 and b_2 are in the same cone then the Minkowski sum of the polytopes $Ax = b_1,\ x \geq 0$ and $Ax = b_2,\ x \geq 0$ is equal to the polytope $Ax = b_1 + b_2,\ x \geq 0$.*

Sketch of proof: We shall use P_b to denote the polytope given by the linear system $Ax = b,\ x \geq 0$. It is easy to show that $P_{b_1} + P_{b_1} \subseteq P_{b_1+b_2}$ and that all three of these polytopes are bounded. Also, b_1 and b_2 being in the same cone implies that b_1, b_2 and $b_1 + b_2$ are all in the same cone (though this is not, in general, a transitive relationship). For $\sigma \in \Gamma$ and $u \in \mathbb{R}^m$ we define $\nu(\sigma, u)$ to be the vector in $\mathbb{R}^n$ such that $\nu(\sigma, u)_i = 0 \; \forall i \notin \sigma$ and $\nu(\sigma, u)_\sigma = A_\sigma^{-1} u$. All vertices of $Ax = b$ are of the form $\nu(\sigma, b)$ (though $\nu(\sigma, b)$ is not always a vertex or even always in P_b).

Let v be an arbitrary vertex of $P_{b_1+b_2}$ and σ be such that $v = \nu(\sigma, b_1 + b_2)$. We note that $\nu(\sigma, b_1) + \nu(\sigma, b_2) = \nu(\sigma, b_1 + b_2)$ and the "in the same cone" relationship guarantees that $\nu(\sigma, b_1) \in P_{b_1}$ and $\nu(\sigma, b_2) \in P_{b_2}$. The arbitrary choice of v completes the proof.

There are no more than $m\binom{n}{m}$ elements in Ψ so the orthant is split into no more than $\left(m\binom{n}{m}\right)^m$ cones. By combining Theorem 1 with Theorem 2 we see that the restriction of the counting function $f_A(b)$ to any of these cones is a polynomial of degree no more than $n - m$.[4, 17, 18]

2.2 Interpolation

We now have a decomposition of the positive orthant into cones such that $f_A(b)$ restricted to any cone is an unknown polynomial of degree $n - m$. It remains to, for a given cone, derive the polynomial. The method we use is to find an integral point y in the interior of the cone. The polynomial for a cone is completely determined by the values of $f_A((n-m)y + \Delta)$ where Δ runs over all $\binom{n}{m}$ nonnegative integral vectors in the simplex S_{n-m} in $\mathbb{Z}^m$ such that $1 \cdot \Delta \leq n - m$. We call the required $f_A((n-m)y + \Delta)$ "small sub-problems" and describe how to solve them in the next section. With the small sub-problem solutions in hand, the interpolation can be done very quickly. We introduce the multivariate "Lagrange polynomial" $l_{b,x,d}(\cdot)$ which allows us to perform the interpolation without having to (computationally) invert a $\binom{n}{m}$ by $\binom{n}{m}$ matrix. In this notation we call $b \in \mathbb{Z}^m$ the "base", $x \in \mathbb{Z}^m$ is such that $x - b \in S_d$ and d is the degree. We define $l_{b,x,d}(\cdot)$ as the unique polynomial map from $\mathbb{Z}^m \to \mathbb{R}$ of degree d such that $l_{b,x,d}(x) = 1$ and $l_{b,y,d}(y) = 0$ when y is integral and $y - b \in S$ and $y \neq x$. These polynomials have a very simple explicit formula:

$$l_{b,x,d}(v) \stackrel{\text{def}}{=} \begin{cases} \prod_{i=1}^{d} (1 + (1 \cdot b - 1 \cdot v)/i) & x = b \\ \frac{(v_i - b_i)}{(x_i - b_i)} l_{b+E^i,x,d-1}(v) & x_i > b_i,\ x_j = b_j \ \forall j < i \end{cases}$$

Then the desired polynomial is:

$$p(v) \stackrel{\text{def}}{=} \sum_{\Delta \in S} f_A((n-m)y + \Delta) l_{(n-m)y,(n-m)y+\Delta,n-m}(v).$$

The sum can be evaluated symbolically with v as an indeterminate vector or can be evaluated very quickly using almost no space if $v \in \mathbb{Z}^m$ is given explicitly.

2.3 Size of Sub-Problems

By Hadamard's inequality[14] the system $Cy \geq 0,\ y \geq 0$ (where the rows of C or elements of Ψ or their opposites) representing an arbitrary cone of the type mentioned above has an integral interior point y such that $y_i \leq m^{m/2+1}$ (remember, because A is totally unimodular all vectors in Ψ are $0, \pm 1$). Graham and Sloane studied the equality version of this problem[10] and Văn H. Vũ of Yale has shown [19] that the $m^{O(m)}$ bound is essentially best possible.

2.4 Divided and Conquer

For small problems (many of those arising from magic square problems) we have devised a specialized decomposition which we call "divide and conquer." This method is similar to that of Chan and Robbins [3] who used it to compute the volume of the order 7 and order 8 Birkhoff polytopes. To count the number of nonnegative integral solutions to $Ax = b$ we divide the columns of A into two sets of size about n_1, n_2 ($n_1 + n_2 = n$, $n_1 = \lfloor n/2 \rfloor$): A_1, A_2 and then explicitly enumerate all possible b_1, b_2 such that $A_1 x_1 = b_1,\ x_1 \geq 0$ and that $A_2 x_2 = b_2,\ x_2 \geq 0$ are feasible and $b_1 + b_2 = b$. We have a recursive expansion of the form:

$$\begin{aligned}&\sharp(\{x|Ax = b,\ x \geq 0,\ x \in \mathbb{Z}^n\}) = \\ &\quad \textstyle\sum_{b_1,b_2\ b_1+b_2=b, b_1,b_2 \in Z^m \geq 0} \quad \sharp(\{x_1|A_1x_1 = b_1,\ x \geq 0,\ x \in \mathbb{Z}^{n_1}\}) \times \\ &\qquad\qquad\qquad\qquad\qquad\qquad \sharp(\{x_2|A_2x_2 = b_2,\ x \geq 0,\ x \in \mathbb{Z}^{n_2}\}).\end{aligned}$$

This divides a m by n counting problem into no more than B^m sub-problems. We combine this with dynamic programming, meaning that we take care to never solve any sub-problem twice. There are no more than $n \log n B^m$ possible sub-problems which is sufficient to yield a runtime of $n \log n B^{2m}$.

2.5 Zero/One Method

Another method for small problems is the "even/odd" decomposition given by:

$$\begin{aligned}&\sharp(\{x|Ax = b,\ x \geq 0,\ x \in \mathbb{Z}^n\}) = \\ &\quad \textstyle\sum_{r \in R_A,\ r \leq b,\ r = b \bmod 2} \quad \sharp(\{z|z \in \{0,1\}^n,\ Az = r\}) \times \\ &\qquad\qquad\qquad\qquad\qquad \sharp(\{y|Ay = (b-r)/2,\ y \geq 0,\ y \in \mathbb{Z}^n\})\end{aligned}$$

where R_A is the set $\{Az | z \in \{0,1\}^n\}$. We apply this decomposition recursively to the non-zero/one problems until we have driven all the entries of b to zero. Again, for efficiency we use a dynamic programming to avoid recalculating the value of any sub-problem. We note that in this recursion a limited number of right hand sides (of the form $(b-r)/2$) are formed. In fact we solve no more than $n^m \log B$ non-zero/one sub-problems.

The zero/one problems themselves can be solved either using the a specialization of the divide and conquer counter to zero/one problems or by another application of dynamic programming using the recursive formula:

$$\begin{aligned}&\sharp(\{z|Az = b,\ z \in \{0,1\}^n\}) = \\ &\quad \sharp(\{z_1|\tilde{A}z_1 = b,\ z_1 \in \{0,1\}^{n-1}\}) + \\ &\quad \sharp(\{z_2|\tilde{A}z_2 = b - A^1,\ z_2 \in \{0,1\}^{n-1}\}).\end{aligned}$$

where A^1 is the first column of A and $\tilde{A}$ is the matrix formed by all columns of A except the first. Either method yields a run time no worse than $2^{O(m \log n)}$.

Combining these observations lets us conclude that the zero/one counter can be implemented in $2^{O(m \log n)} \log B$ time. Given our problem size bounds above this is sufficient to run the entire interpolation method in time $2^{O(m \log n)}$.

3 Contingency Tables

A special problem is to find the number of non-negative matrices that sum to given row and column totals. Such matrices are called contingency tables.[6] If the matrices are square and all rows and columns are constrained to sum to the same value then the matrices are called "magic squares".[15] As mentioned before counting contingency tables is $\sharp P$ hard. Counting or uniformly generating contingency tables has applications in statistics.[5]

The generating function for $h \times w$ contingency tables is the following:

$$\prod_{i=1\cdots h,\ j=1\cdots w} \frac{1}{1 - x_i y_j}$$

Even for magic squares ($h = w, B =$ each row/column total, $m = h + w - 1$, $n = (h-1)(w-1)$) the counting can be troublesome (though no complexity result is known). Complete expansion of the generating function can require B^{2h} space and similar time. Dynamic programming methods[9] can perform the calculation using hB^{h-1} space. This still yields a rough runtime of B^{2h} as each entry in the dynamic programming table depends on $O(B^h)$ previous entries. Young/Ferris diagram based methods [12] roughly need to store all Young/Ferris diagrams on hB dots with $\leq h$ rows (yielding $\approx B^{h-1}$ storage). Once B is as big as order h^2 it can be shown that there are at least $(B/(2h))^{h-1}$ diagrams that are connected to roughly $(B/(2h))^{h-1}$ diagrams each yielding $\approx (B/2h)^{2h-2}$ work.

The divide and conquer approach, mentioned above, can be specialized to this problem and compares favorably to these methods, if performed with care. If A is the matrix representing the contingency table counting problem, one can always split the columns of A approximately in half in such a way that both the sub-problems are contingency table problems with either the width or height halved. By combining with the dynamic programming technique, we get a runtime of $B^{3h/2}$ using B^h storage.

This runtime is noticed by computing the work done at each stage of a dynamic programming implementation of divide and conquer. The following table shows how many problems occur at each level, what their width and height is, how many subproblems each problem needs from the next level, and how much work is done at each level (nodes times connections).

level	1	2	3	4	5	6	$\cdots$
nodes	1	B^h	B^h	$B^{\frac{3}{4}h}$	$B^{\frac{1}{2}h}$	$B^{\frac{3}{8}h}$	
width	h	h	$h/2$	$h/2$	$h/4$	$h/4$	
height	h	$h/2$	$h/2$	$h/4$	$h/4$	$h/8$	
edges/node	B^h	$B^{h/2}$	$B^{h/2}$	$B^{h/4}$	$B^{h/4}$	$B^{h/8}$	
work	B^h	$B^{\frac{3}{2}h}$	$B^{\frac{3}{2}h}$	B^h	$B^{\frac{3}{4}h}$	$B^{\frac{1}{2}h}$	

For $B \in o(h^{8/3})$ the divide and conquer counter is faster than the even/odd counter and for $B \in o(h^{4/3})$ the divide and conquer counter takes fewer steps than there are terms in the multivariate counting polynomial.

4 Computational Results

4.1 zero/one results

The following single 5×4 problem was solved in 20 minutes using the zero one method (a previous solution by the interpolation method took just over one week): There are exactly 23196436596128897574829611531938753 ($\approx$ 2.3 * 10^{34}) non-negative 5 by 4 integer matrices whose rows sum to $[\ 182,\ 778,\ 3635,\ 9558,\ 11110\ ]$ and columns sum to $[\ 3046,\ 5173,\ 6116,\ 10928\ ]$.

4.2 4 by 4 contingency tables

The 4×4 contingency table problem splits into (after some symmetries are removed) 3694 cones such that the counting function restricts to a degree 9 polynomial in 7 variables in each cone. This means each polynomial is determined by 11440 coefficients so can be interpolated from 11440 sufficiently general evaluations. Each of the 3694 tasks seems to represent about 3 hours of Pmax CPU time (using the divide and conquer counter), so the total job represented about 1.5 Pmax CPU years. This task was completed in just over 6 weeks using Peter Stout's **WAX** [16] system which effectively simulates a coarse-grain parallel supercomputer (employing the numerous idle workstations at CMU). This polynomial was stored on disk and was able to solve typical 4×4 contingency tables (with margins around a couple of hundred) in 3 seconds. This was made available on the World Wide Web in the form of an online calculator.

4.3 Magic Squares

We know [15] that the number of non-negative integral points in the $n \times n$ Birkhoff polytope (i.e. the number of magic squares) is a polynomial of degree $(n-1)^2$ in r (r an integer). Denote this polytope by $H_n(r)$. In fact, the stronger result is known that:

$$\sum_{r \geq 0} H_n(r)\lambda^r = \frac{h_0 + h_1\lambda + \cdots h_d\lambda^d}{(1-\lambda)^{(n-1)^2+1}}$$

where $d = n^2 - 3n + 2$ and the h_i are non-negative integers such that $h_i = h_{d-i},\ i = 0, 1, \cdots d$.

This means that $H_7(r)$ is completely determined by the values of $H_7(1)$ through $H_7(15)$ and $H_8(r)$ is completely determined by the values of $H_8(1)$ through $H_8(21)$. Other authors have reported similar results [3]. These values were computed by the divide and conquer technique and are tabulated below:

r	$H_7(r)$	$H_8(r)$
1	5040	40320
2	9135630	545007960
3	4662857360	1579060246400
4	936670590450	1455918295922650
5	94161778046406	569304690994400256
6	5562418293759978	114601242382721619224
7	215717608046511873	13590707419428422843904
8	5945968652327831925	1046591482728407939338275
9	123538613356253145400	56272722406349235035916800
10	2023270039486328373811	2233160342369825596702148720
11	27046306550096288483238	68316292103293669997188919040
12	303378141987182515342992	1667932098862773837734823042196
13	2920054336492521720572276	33427469280977307618866364694400
14	24563127009195223721952590	562798805673342016752366344185200
15	183343273080700916973016745	8115208977465404874100226492575360
16		101857066150530294146428615917957029
17		1128282526405022554049557329097252992
18		11161302946841260178530673680176000200
19		99613494890126594335550124219924540800
20		809256770610540675454657517194018680846
21		6031107989875562751266116901999327710720

With these valuations it is trivial to determine both the polynomials $H_7(r)$, $H_8(r)$ and all the coefficients in the numerators of the generating functions given above. To derive the polynomials we use $H_n(0) = 1$, $H_n(-1) = H_n(-2) = \cdots H_n(-n+1) = 0$ and $H_n(-n-r) = (-1)^{n-1} H_n(r)$. Note that the data given here for $H_7(r)$ and $H_8(r)$ obey conjecture (v) on page 26 of [15].

References

[1] A. BARVINOK, *A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed*, in Symposium on Foundations of Computer Science, no. 34, IEEE Computer Society Press, November 1993.

[2] A. I. BARVINOK, *Computing the Ehrhart polynomial of convex lattice polytope*, Discrete Comput. Geom., (1994), pp. 35–48.

[3] C. S. CHAN AND D. P. ROBBINS, *On the volume of the polytope of doubly stochastic matrices, eprint: math.co/9806076*, 1998.

[4] W. DAHMEN AND C. MICCHELLI, *The number of solutions to linear Diophantine equations and multivariate splines*, Transactions Amer. Math. Soc., (1988), pp. 509–532.

[5] P. DIACONIS AND B. EFRON, *Testing for independence in a two-way table: New interpretations of the chi-square statistic*, Annals of Statistics, 13 (1985), pp. 845–874.

[6] P. DIACONIS AND A. GANGOLLI, *Rectangular arrays with fixed margins*. to appear: Proceedings of the workshop on Markov Chains, Institute of Mathematics and its Applications, University of Minnesota, April 1994.

[7] M. DYER, R. KANNAN, AND J. MOUNT, *Sampling contingency tables*, Random Struct. Alg., 10 (1997), pp. 487–506.

[8] W. FULTON, *Introduction to Toric Varieties*, Princeton University Press, 1993.

[9] M. GAIL AND N. MANTEL, *Counting the number of $r \times c$ contingency tables with fixed margins*, Jour. Amer. Statist. Assoc., (1977), pp. 859–862.

[10] R. L. GRAHAM AND N. J. A. SLOANE, *Anti-Hadamard matrices*, Linear Algebra and its Applications, (1984), pp. 113–137.

[11] P. GRUBER AND C. LEKKERKERKER, *Geometry of Numbers*, North-Holland, second edition ed., 1987.

[12] D. M. JACKSON AND H. J. VAN REES, *The enumeration of generalized double stochastic nonnegative integer square matrices*, SIAM J. Comput., 4 (1975), pp. 474–477.

[13] J. MOUNT, *Application of Convex Sampling to Optimization and Contingency Table Generation/Counting*, PhD thesis, Carnegie Mellon, May 1995. t.r. number CMU-CS-95-152.

[14] A. SCHRIJVER, *Theory of Linear and Integer Programming*, Wiley, 1986.

[15] R. P. STANLEY, *Combinatorics and Commutative Algebra*, Birkhäuser, 2nd ed., 1996.

[16] P. STOUT, *Wax: A Wide Area Computation System*, PhD thesis, Carnegie-Mellon Univ, December 1994. t.r. number CMU-CS-94-230.

[17] B. STURMFELS, *How to count integer matrices*. preprint, December 1993.

[18] ——, *On vector partition functions*, Journal of Combinatorial Theory, Series A, 72 (1995), pp. 302–309.

[19] V. H. VŨ, *Ill-conditioned $(0,1)$ and $(-1,1)$ matrices and their applications*. manuscript, 1996.