Programming and Computer Software, Vol. 30, No. 2, 2004, pp. 105–109. Translated from Programmirovanie, Vol. 30, No. 2, 2004.

Fast Multiplication and Sparse Structures

G. I. Malaschonok and E. S. Satina
Institute of Mathematics, Physics, and Informatics, Derzhavin State University, Tambov, ul. Internatsional'naya 33, 392622 Russia
e-mail: malaschonok@math-iu.tstu.ru
Received June 16, 2003

Abstract—The applicability of fast multiplication algorithms to sparse structures is discussed. Estimates for the degree of sparseness of matrices and polynomials are given for which fast multiplication algorithms have advantages over standard multiplication algorithms in terms of the multiplicative complexity. Specifically, the Karatsuba and Strassen algorithms are studied under the assumption of the uniform distribution of zero elements.

1. INTRODUCTION

The use of fast algorithms for the multiplication of matrices and polynomials is limited to a certain range of problems. In practice, it is important to understand whether one or another fast algorithm is efficient for solving a given problem.

In this work, we study the efficiency of the Karatsuba [1] and Strassen [2] algorithms for the multiplication of sparse polynomials and matrices. Specifically, we study how the efficiency of the algorithms depends on the density coefficient ρ, which is the ratio of the number of nonzero coefficients to the total number of matrix elements or coefficients of the polynomial. If $\rho = 1$, all coefficients are nonzero, and the fast algorithm is more efficient than the standard one. As ρ decreases, the efficiency of the fast algorithm diminishes, and, at a certain value of ρ, the multiplicative complexity of the fast algorithm becomes equal to that of the standard one. The problem is to determine this (threshold) value of ρ for the given fast algorithm. We will compare algorithms in terms of the multiplicative complexity, which is defined as the number of multiplications of the coefficients.

Consider a standard algorithm for the multiplication of polynomials $f(x)$ and $g(x)$. The numbers of nonzero coefficients in these polynomials are equal to $(\deg(f) + 1)\rho(f)$ and $(\deg(g) + 1)\rho(g)$, respectively. Hence, the total number of multiplications of the coefficients is

$$C^p_{Stand} = (\deg f + 1)(\deg g + 1)\rho(f)\rho(g). \tag{1}$$

The Karatsuba algorithm for the multiplication of polynomials of degree $2n - 1$ consists in the recursive computation by the formula

$$\begin{aligned} fg &= (a + bx^n)(c + dx^n) \\ &= ac + (ac + bd - (a - b)(c - d))x^n + bdx^{2n}, \end{aligned} \tag{2}$$

where a, b, c, and d are polynomials of degree $n - 1$. This formula contains only three operations of the multiplication of polynomials of degree $n - 1$.

We will estimate the complexity of the Karatsuba algorithm assuming that $n = 2^N - 1$. In this case, the number of the multiplication operations is

$$C^K_N = 3^N. \tag{3}$$

If, in the course of the computation by the Karatsuba algorithm, the sparseness of the polynomials is not taken into account, we can find the critical density for $n = 2^N - 1$ by equating the right-hand sides of Eqs. (1) and (3),

$$\rho_K = \sqrt{\rho(f)}\sqrt{\rho(g)} = (3/4)^{N/2}. \tag{4}$$

Hence, the complexity of the Karatsuba algorithm is less than that of the standard algorithm if $\rho > \rho_K$.

Now, we consider a standard algorithm for the multiplication of square matrices. Let A and B be matrices of order n, and let $\rho(A)$ and $\rho(B)$ be densities of the matrices. Assuming that nonzero elements are distributed uniformly, the total number of the multiplications of nonzero elements is found to be

$$C_{Stand} = n^3\rho(A)\rho(B). \tag{5}$$

The multiplication of two matrices of order $n = 2^N$ by the Strassen algorithm requires

$$C^S_N = 7^N \tag{6}$$

multiplications of the matrix elements. If, in the course of the computation by the Strassen algorithm, the sparseness of the matrices is not taken into account, we can find the critical density for $n = 2^N$ by equating the right-hand sides of Eqs. (5) and (6),

$$\rho_S = \sqrt{\rho(A)}\sqrt{\rho(B)} = (7/8)^{N/2}. \tag{7}$$

Hence, the complexity of the Strassen algorithm is less than that of the standard algorithm if $\rho > \rho_S$.

Table 1. Critical values of density for the Karatsuba (ρ_K) and Strassen (ρ_S) algorithms that do not take into account the presence of zero elements

n	2	4	8	16	32
ρ_K	0.8660	0.7500	0.6495	0.5625	0.4871
ρ_S	0.9354	0.8750	0.8185	0.7652	0.7162
n	64	128	256	512	1024
ρ_K	0.4219	0.3654	0.3166	0.2740	0.2375
ρ_S	0.6700	0.6267	0.5862	0.5483	0.5129
n	2048	4096	8192	16384	32768
ρ_K	0.2055	0.1780	0.1541	0.1335	0.1156
ρ_S	0.4798	0.4488	0.4198	0.3927	0.3673

Several critical values of ρ_S and ρ_K are given in Table 1.

These estimates are, in fact, rather high. Of interest are more accurate estimates of the critical density values, which are based on more elaborate estimates of the complexity of fast algorithms for sparse data. This problem is dealt with in the remaining part of the paper.

2. COMPLEXITY ESTIMATE FOR THE KARATSUBA ALGORITHM FOR THE MULTIPLICATION OF SPARSE POLYNOMIALS

To take into account the sparseness of the polynomials in the Karatsuba algorithm, we assume that the recursive formula (2) is used, with the recursion depth being equal to k. The product of polynomials of degree 2^{N-k} is computed in a standard way. The multiplication of polynomials f and g requires

$$(\deg f + 1)(\deg g + 1)\rho(f)\rho(g)$$

multiplications of the coefficients. If $k = N$ and the recursion terminates on polynomials of zero degree with the average density ρ, then the average number of nonzero products is ρ^2.

Consider the algorithm of the computation of the product by formula (2). We assume that the degrees of the cofactors are equal to $2^N - 1$ and the nonzero elements are distributed uniformly with the density a. Then, the first recursive computation requires three multiplications of polynomials of degree $2^{N-1} - 1$; in two cases, the cofactor densities are equal to a, and, in one case, the density is $\phi(a) = 2a - a^2$. Note that $\phi(a)$ is the density of the sum of the polynomials in the case where a is the density of the addends and the field of the coefficients has zero characteristic. Note also that, for a finite field containing p elements, $\phi(a) = 2a - a^2\dfrac{p}{p-1}$.

If the recursion depth is equal to two, we have nine products of polynomials of degree $2^{N-2} - 1$; in four cases, the density is equal to a; in four other cases, the density is $\phi(a)$; and, in one case, $\phi^2(a) = \phi(\phi(a))$.

Let the recursion depth be k, and let the degrees of polynomials be $2^{N-2} - 1$. Denote by $a_{k,j}$ the number of products of polynomials of degree $\phi^j(a)$. Let us find the generating function

$$F_k(z) = \sum_{j=0}^{k} a_{k,j} z^j$$

for the sequence $a_{k,j}$. Taking into account that

$$F_0(z) = 1, \quad F_k(z) = (2+z)F_{k-1}(z), \quad k = 1, 2, \ldots,$$

we obtain

$$F_k(z) = (2+z)^k = \sum_{j=0}^{k} \binom{k}{j} 2^{k-j} z^j,$$

$$a_{k,j} = \binom{k}{j} 2^{k-j}.$$

If the recursion depth is k and the number of the multiplications of the nonzero coefficients is estimated as in the case of the standard multiplication, we obtain the following estimate for the complexity of the recursive algorithm:

$$C_k^p = \sum_{j=0}^{k} \binom{k}{j} 2^{k-j} (\phi^j(a))^2. \tag{8}$$

Here, the function

$$\varphi^j(a) = \sum_{s}^{j} \binom{j}{s} (-1)^s 2^{j-s} a^{2^s} \tag{9}$$

is obtained by applying j times the operator $(2 - \tau)$, where τ is the operator of raising to the second power,

$$\phi(a) = (2-\tau)(a), \quad \phi^j(a) = (2-\tau)^j(a).$$

Results of the comparison of the complexity estimate obtained with the complexity $a^2 2^N$ of the standard multiplication for different recursion depths (1, 2, …, 15) and different values of density of the source polynomials are presented in Table 2.

The left column shows the recursion depth, and the upper row, the ratio of the complexity of the Karatsuba algorithm to that of the standard multiplication. Numbers in the cells show the corresponding density of the polynomial cofactors.

For example, for the recursion depth equal to 15, the multiplicative complexity of the Karatsuba algorithm is two times less than that of the standard algorithm when the density of the polynomials is 0.1503. If the density of the polynomials is less than 0.1 and the degrees of the polynomials do not exceed 2^{15}, the complexity of the Karatsuba algorithm is greater than that of the standard algorithm.

Note that, if the field of the coefficients is finite and contains p elements, formula (9) takes the form

$$\phi^j(a) = \sum_s^j \binom{j}{s}\left(-\frac{p}{p-1}\right)^s 2^{j-s} a^{2^s}.$$

3. COMPLEXITY ESTIMATE FOR THE STRASSEN ALGORITHM FOR THE MULTIPLICATION OF SPARSE MATRICES

The Strassen formula for the multiplication of square matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ of order two is as follows:

$$AB = \begin{pmatrix} t_1 + t_4 - t_5 + t_7 & t_3 + t_5 \\ t_3 + t_4 & t_1 + t_3 - t_2 + t_6 \end{pmatrix}, \qquad (10)$$

where

$$t_1 = (a_{11} + a_{22})(b_{11} + b_{22}),$$
$$t_2 = (a_{21} + a_{22})b_{11},$$
$$t_3 = a_{11}(b_{12} - b_{22}),$$
$$t_4 = a_{22}(b_{21} - b_{11}),$$
$$t_5 = (a_{11} + a_{12})b_{22},$$
$$t_6 = (a_{21} - a_{11})(b_{11} + b_{12}),$$
$$t_7 = (a_{11} - a_{22})(b_{21} + b_{22}).$$

Applying it recursively, we obtain an algorithm for the multiplication of matrices of order $n = 2^s$ with $n^{\log_2 7}$ ($\approx n^{2.807}$) multiplication operations.

To take into account sparseness of matrices in the Strassen algorithm, we assume that the recursive algorithm (10) is implemented with the recursion depth k and that the matrices of order $n = 2^{N-k}$ are multiplied in a standard way. To multiply matrices of order $n = 2^{N-k}$ with densities ρ_1 and ρ_2, $n^3\rho_1\rho_2$ multiplication operations are required. If $k = N$ and the recursion terminates on matrices of order one with the average density ρ, then the average number of nonzero products is ρ^2.

Consider the algorithm of the multiplication of matrices based on formula (10). We assume that the matrices A and B are of order 2^N and that the nonzero elements are distributed uniformly with the density a. Then, the first recursive computation involves seven multiplications of matrices of order 2^{N-1}; in four cases, the cofactor densities are equal to a and $\phi(a) = 2a - a^2$, and, in three cases, the density of both cofactors is $\phi(a)$.

If the recursion depth is equal to two, we have 49 products of matrices of order 2^{N-2}; in 4×2 cases, the

Table 2. Densities of polynomials for the given recursion depth (k) and the given ratio of the complexity of the Karatsuba algorithm to that of the standard algorithm

k	1	1/2	1/4
$k = 1$	0.5858		
$k = 2$	0.5236		
$k = 3$	0.4668	0.8819	
$k = 4$	0.4150	0.7344	
$k = 5$	0.3680	0.6237	0.9702
$k = 6$	0.3256	0.5353	0.8247
$k = 7$	0.2875	0.4621	0.7054
$k = 8$	0.2534	0.4002	0.6061
$k = 9$	0.2229	0.3473	0.5224
$k = 10$	0.1959	0.3018	0.4512
$k = 11$	0.1719	0.2624	0.3903
$k = 12$	0.1507	0.2283	0.3379
$k = 13$	0.1320	0.1985	0.2928
$k = 14$	0.1155	0.1727	0.2539
$k = 15$	0.1009	0.1503	0.2202

k	1/8	1/16	1/32	1/64
$k = 7$	1.0			
$k = 8$	0.8895			
$k = 9$	0.7654	1.		
$k = 10$	0.6597	0.9482		
$k = 11$	0.5694	0.8186		
$k = 12$	0.4920	0.7072	1.0	
$k = 13$	0.4254	0.6112	0.8709	
$k = 14$	0.3681	0.5285	0.7532	1.
$k = 15$	0.3187	0.4572	0.6516	0.9246

Table 3. Densities of matrices for the given recursion depth (k) and the given ratio of the complexity of the Strassen algorithm to that of the standard algorithm

	1	1/2	1/4	1/8
$k = 1$	2/3			
$k = 2$	0.6637	1.0		
$k = 3$	0.6345	0.9272		
$k = 4$	0.5956	0.8555		
$k = 5$	0.5564	0.7928		
$k = 6$	0.5187	0.7363		
$k = 7$	0.4833	0.6847	0.969	
$k = 8$	0.4503	0.6374	0.9016	
$k = 9$	0.4196	0.5938	0.8398	
$k = 10$	0.3912	0.5534	0.7827	
$k = 11$	0.3649	0.5160	0.7298	
$k = 12$	0.3404	0.4813	0.6808	0.9627
$k = 13$	0.3176	0.4491	0.6352	0.8983
$k = 14$	0.2964	0.4192	0.5928	0.8384
$k = 15$	0.2767	0.3913	0.5535	0.7826

	1	1/2	1/4	1/8
$k = 1$	2/3			
$k = 2$	0.6637	1.0		
$k = 3$	0.6345	0.9272		
$k = 4$	0.5956	0.8555		
$k = 5$	0.5564	0.7928		
$k = 6$	0.5187	0.7363		
$k = 7$	0.4833	0.6847	0.969	
$k = 8$	0.4503	0.6374	0.9016	
$k = 9$	0.4196	0.5938	0.8398	
$k = 10$	0.3912	0.5534	0.7827	
$k = 11$	0.3649	0.5160	0.7298	
$k = 12$	0.3404	0.4813	0.6808	0.9627
$k = 13$	0.3176	0.4491	0.6352	0.8983
$k = 14$	0.2964	0.4192	0.5928	0.8384
$k = 15$	0.2767	0.3913	0.5535	0.7826

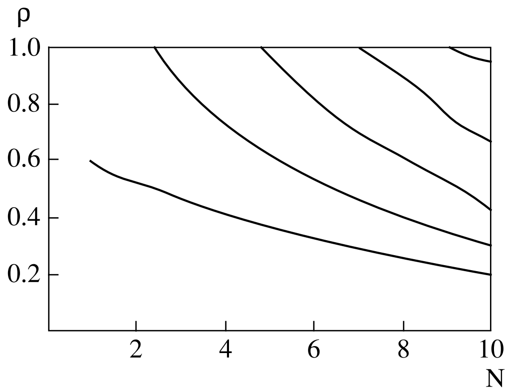

Fig. 1.

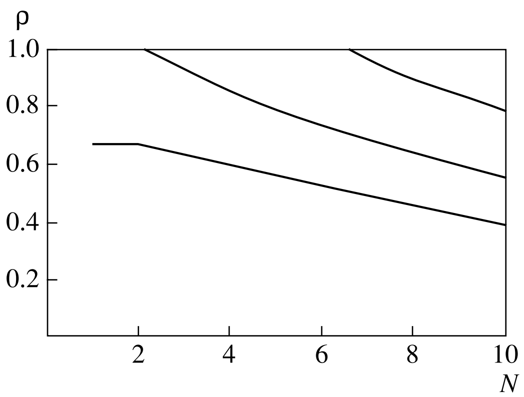

Fig. 2.

densities are equal to a and $\phi^2(a)$; in 4×2 cases, the densities of both cofactors are equal to $\phi(a)$; in $4 \times 3 + 3 \times 4$ cases, $\phi(a)$ and $\phi^2(a)$; and, finally, in 3×3 cases, the density of both cofactors is $\phi^2(a)$.

Let the recursion depth be k, and let the orders of matrices be 2^{N-k}. Denote by $a^k_{i,j}$ the number of products of matrices that have densities $\phi^i(a)$ and $\phi^j(a)$. Let us find the generating function

$$F_k(y, z) = \sum_{i=0}^{k}\sum_{j=0}^{k} a^k_{i,j} y^i z^j$$

for the sequence $a^k_{i,j}$. Taking into account that

$$F_0(y, z) = 1, \; F_k(z) = (2y + 2z + 3yz)F_{k-1}(y, z), \quad k = 1, 2, \ldots,$$

we obtain the equation

$$F_k(y, z) = (2y + 2z + 3yz)^k = \sum_{i=0}^{k}\sum_{j=0}^{i}\binom{k}{i}\binom{i}{j} 2^{k-j} 3^j y^{k-i+j} z^i,$$

from which we find the desired quantities $a^k_{i,j}$,

$$a^k_{i,j} = \binom{k}{j}\binom{j}{i+j-k} 2^{2k-i-j} 3^{i+j-k}.$$

If the recursion depth is k and the number of the multiplications of the nonzero coefficients is estimated as in the case of the standard multiplication, we obtain the following estimate for the complexity of the recursive algorithm:

$$C^m_k = \sum_{i=0}^{k}\sum_{j=0}^{i}\binom{k}{j}\binom{j}{i+j-k} 2^{2k-i-j} 3^{i+j-k} \times (2^{N-k})^3 \phi^i(a)\phi^j(a), \tag{11}$$

where, for the field of coefficients of zero characteristic, $\phi^j(a)$ are defined in (9).

Results of the comparison of the complexity estimate obtained with the complexity of the standard multiplication $a^2(2^N)^3$ for different recursion depths (1, 2, …, 15) and different values of density of the source matrices are presented in Table 3.

The left column shows the recursion depth, and the upper row, the ratio of the complexity of the Strassen algorithm to that of the standard multiplication. Numbers in the cells show the corresponding density of the matrix cofactors.

If the density of matrices is greater than 2/3, the Strassen algorithm is more efficient than the standard algorithm for any recursion depth. If the density of the matrices is less than 0.39 and the order of matrices is less than 1000, the Strassen algorithm is less efficient than the standard algorithm.

4. CONCLUSIONS

The results of the above discussions can be summarized in several practical recommendations, for example, for polynomials whose degree is not greater than 1000. If the density of the cofactors is greater than 0.59, the Karatsuba algorithm is always more advantageous. If the density of the cofactors is less than 0.20, the standard algorithm is more efficient than the Karatsuba algorithm. In practice, one can take advantage of the plot depicted in Fig. 1.

This plot shows the contour lines (1, 1/2, 1/4, 1/8, 1/16) of the ratio of the complexity of the Karatsuba algorithm to that of the standard algorithm. The horizontal axis presents the logarithm $N = \log_2 n$ of the degree of the polynomial cofactors, and the vertical axis shows the density of the polynomials.

As to the Strassen algorithm, for matrices of the order less than 1000, it is more efficient than the standard algorithm when the matrix density is greater than 0.66. If the density is less than 0.39, the standard algorithm is more efficient. In practice, one can take advantage of the plot depicted in Fig. 2.

This plot shows the contour lines (1, 1/2, 1/4) of the ratio of the complexity of the Strassen algorithm to that of the standard algorithm. The horizontal axis presents the logarithm $N = \log_2 n$ of the matrix order, and the vertical axis corresponds to the density of the matrices.

Note that the Vinograd algorithm [3], which has the same number of multiplications but less additions, is less efficient than the Strassen algorithm in the case of sparse matrices. In this algorithm, each recursion requires the summation of four submatrices rather than two submatrices, as in the Strassen algorithm. Hence, the density of the operands grows faster.

ACKNOWLEDGMENTS

The work was supported in part by the Russian Ministry of Education, project no. E02-2.0-98, and by the Human Capital Foundation, project no. 23-03-24.

REFERENCES

1. Karatsuba, A., *Dokl. Akad. Nauk SSSR*, 1962, vol. 145, pp. 293–294.
2. Strassen, V., Gaussian Elimination Is Not Optimal, *Numerische Mathematik*, 1969, vol. 13, pp. 354–356.
3. Knuth, D.E., *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms*, Reading: Addison-Wesley, 1969. Translated under the title *Iskusstvo programmirovaniya dlya EVM*, tom 2: *Poluchislennye algoritmy*, Moscow: Mir, 1977.

Spell: Ok