INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

4

STUDIES IN

ALGORITHMS FOR FAST RECTANGULAR MATRIX MULTIPLICATIONS AND THEIR APPLICATIONS

by

XIAOHAN HUANG

A dissertation submitted to the Graduate Faculty in Mathematics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1997

This manuscript has been read and accepted for the Graduate Faculty in Mathematics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

6/3/97 — Victor Pan
Date — Chair of Examining Committee

6/5/97 — I. Chavel (ATV)
Date — Executive Officer

MICHAEL ANSHEL

MYONG-HI KIM

VICTOR Y. PAN

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

STUDIES IN ALGORITHMS FOR FAST RECTANGULAR MATRIX MULTIPLICATIONS AND THEIR APPLICATIONS

by

Xiaohan Huang

Advisor: Professor Victor Y. Pan

The first part of the dissertation describes the studies and results in developing the algorithms for fast rectangular matrix multiplications. In this part, we present two algorithms for the problem $< n, n, n^2 >$ of computing the product of an $n \times n$ matrix by an $n \times n^2$ matrix, supporting the asymptotic arithmetic complexity bounds $O(n^\omega)$ with the exponents $3.33984\cdots$ and $3.333953\cdots$, respectively, which are better (less) than previously known record exponent $3.375477\cdots$ by 0.036 and 0.042 respectively. Then, we present fast algorithms for rectangular matrix multiplications for matrix pairs of arbitrary dimensions and deal with the algorithms as the functions of the dimensions. Finally, we have discussed the optimization of the algorithms for rectangular matrix multiplications. Under "majority" situations, our algorithms achieve better matrix exponents than the known ones.

The second part of the dissertation describes the applications of our results of the first part on two topics. In particular, we improve the processor efficiency of fast parallel evaluation of the determinant, the characteristic polynomial and the inversion of a given square matrix, that is, we improve the processor complexity of the problem from $O(n^{2.851})$ to $O(n^{2.837})$. In another application, we have yielded the acceleration of univariate polynomial factorization over a finite field.

Contents

Part I

Algorithms for Fast Rectangular Matrix Multiplications

1 Introduction

Acceleration of matrix multiplication is a major subject of theory and practice of computing (see [Pan], [Pan,a], [GL89/96]). In this dissertation, we study some theoretical aspects of this problem, that is, we improve the known upper estimates for the asymptotic complexity of multiplying rectangular matrices of large sizes. We show further applications of our results to the improvement of the known upper bounds on the deterministic asymptotic parallel complexity of some of the most fundamental matrix computations, such as the evaluation of the determinant, the inverse, and the charateristic polynomial of an $n \times n$ matrix and solving a nonsingular system of linear equations and to the improvement of the known algorithms and complexity estimates for the computation of the polynomials of the composition and the known estimates for the computational complexity of the factorization of univariate polynomial over finite fields, which is a major problem of algebraic computing.

Asymptotic arithmetic complexity of square $n \times n$ matrix multiplication has been studied very extensively and intensively (see e.g. [St69], [Sc81], [CW82], [Pan], [Pan,a], [St86], [St87], [St88], [CW90]). So far, this study has culminated in the record upper bound $O(n^{\omega})$, $\omega < 2.376$ (in terms of the number of arithmetic operations involved) [CW90], which marks a long but still uncompleted way from the classical $\omega = 3$ (before 1969) down towards the best lower bound 2.

Less attention has been paid so far to the complexity of rectangular matrix multiplication, where the most important results are [BD76], [Co82] and [Co96].

The asymptotic complexity of $m \times n$ by $n \times p$ matrix multiplication can be expressed in terms of $A(m, n, p)$ denoting the minimum number of arithmetic operations involved. There is a good motivation, however, to confine the study to bilinear algorithms, which consist of linear operations (additions and multiplications by scalars) and bilinear multiplications

(also called nonscalar or essential multiplications), which multiply pairs of linear forms in the entries of the two input matrices (see more details in the next section). The minimum number $M(m,n,p)$ of bilinear multiplications used in all bilinear algorithms for $m \times n$ by $n \times p$ matrix multiplication is an appropriate measure for the asymptotic complexity due to the following known bound (cf. e.g. [Pan]):

$$A(m^h, n^h, p^h) = O((M(m,n,p))^h) \text{ as } h \to \infty. \tag{1.1}$$

Besides, presently and historically, all the known algorithms supporting the record asymptotic complexity estimates for matrix multiplication have been devised as bilinear algorithms.

We have the following simple known estimates (cf. e.g. [Pan]):

$$M(m,n,1) = mn\ , \tag{1.2}$$

$$M(m,n,p) \le M(m/q, n/q, p/q)M(q,q,q) \tag{1.3}$$

for any q that divides m, n, and p. Furthermore, we have

$$M(m,n,p) = M(m,p,n) = M(n,p,m) = M(n,m,p) = M(p,n,m) = M(p,m,n) \quad \text{[Pan72]}, \tag{1.4}$$

$$M(n,n,r(n)) = n^2 + o(n) \quad \text{if} \quad r(n) = o(\log n),\ n \to \infty \qquad \text{[BD76]}\ ,$$

$$A(n,n,n^r) = O(n^{2+\epsilon}) \text{ for any } \epsilon > 0 \text{ if } r \le 0.197,\ n \to \infty \qquad \text{[Co82]}\ ,$$

$$A(n,n,n^r) = O(n^{2+\epsilon}) \text{ for any } \epsilon > 0 \text{ if } r \le 0.294,\ n \to \infty \qquad \text{[Co96]}\ .$$

By extending (1.3), we obtain that

$$M(m,n,p) = O(q^\omega) \max(mn, np, pm)/q^2, \quad q = \min(m,n,p) \to \infty\ ,$$

provided that $M(q,q,q) = O(q^\omega)$. The latter bound asks for improvement for larger q. The goal of this paper is to extend the techniques and the estimates of the paper [CW90], so as to

obtain such an improvement. Having achieved this goal, we will show the resulting improved estimates for the deterministic parallel asymptotic complexity of the cited fundamental matrix computations, as well as of many other matrix computations reducible to them (cf. [BP94]). Our resulting complexity estimates are theoretical: they only apply to very large input matrices; furthermore, we will also in detail show the acceleration of polynomial factorization over finite fields such as the problems of composition of polynomial, polynomial factorization over a finite field, and Fast Black Box Berlekamp Algorithm.

We organize our presentation as follows. In section 2, we recall some basic concepts, definitions and results on matrix multiplicatins. In sections 3 and 4, we modify slightly the technique of section 6 of [CW90], which gives us an algorithm for $< n, n, n^2 >$ having complexity $O(n^{3.3399})$. This will be a basic pattern for our further study. In section 5, we extend the technique of section 7 of [CW90], improve our algorithm for $< n, n, n^2 >$ and yield the bound $O(n^{3.33396})$. In section 6, we show a basic algorithm for $< n^t, n, n^r >$ for an arbitrary pair of non-negative rational numbers t and r. In section 7, we present an improved algorithm for $< n^t, n, n^r >$ for an arbitrary pair of non-negative rational numbers t and r. In section 8, we compare the algorithms developed in our paper with various other effective algorithms and optimize the process of combining all these old and new algorithms together. In section 9, we extend our improvement of rectangular matrix multiplication to the improvement of the known upper estimates for the parallel complexity of matrix computations. In section 10, we extend our improvement of rectangular matrix multiplication to the improvement of the known upper estimates for the univariate polynomial factorization over finite fields.

2 Definitions and Some Background.

In this section, we will introduce some basic concepts and definitions concerning matrix multiplication, define some new concepts and definitions we need to use in this paper, and recall some basic results.

The problem of multiplying an $m \times n$ matrix by an $n \times p$ matrix (so as to produce an $m \times p$ matrix) will be denoted $< m, n, p >$. Indices i, j, k have ranges from 0 to $m-1$, $n-1$, $p-1$ respectively.

Definition 2.1, *bilinear algorithms for matrix multiplication.* Given a pair of $m \times n$ and $n \times p$ matrices $X = [x_{i,j}]$, $Y = [y_{j,k}]$, compute XY in the following order: First evaluate the linear forms in the x-variables and in the y-variables,

$$L_q = \sum_{i,j} f_{ijq} x_{ij} , \qquad L'_q = \sum_{j,k} f^*_{jkq} y_{jk} , \tag{2.1}$$

then the products $P_q = L_q L'_q$ for $q = 0, 1, \cdots, M-1$, and finally the entries $\sum_j x_{ij} y_{jk}$ of XY, as the linear combinations

$$\sum_j x_{ij} y_{jk} = \sum_{q=0}^{M-1} f^{**}_{kiq} L_q L'_q , \tag{2.2}$$

where f_{ijq}, f^*_{jkq} and f^{**}_{kiq} are constants such that (2.1) and (2.2) are the identities in the indeterminates x_{ij}, y_{jk}, for $i = 0, 1, \cdots, m-1$; $j = 0, 1, \cdots, n-1$; $k = 0, 1, \cdots, p-1$. M, the total number of all multiplications of L_q by L'_q, is called the *rank of the algorithm*, and the multiplications of L_q by L'_q are called the *bilinear steps* of the algorithm or *bilinear multiplications.*

The notation $L \rightarrow < m, n, p >$ indicates the existence of a bilinear algorithm requiring L essential (bilinear) multiplications in order to compute the indicated matrix product.

If the algorithm is an "any precision approximation (APA) algorithm" [BCLR], we write $L \xrightarrow{\lambda} < m,n,p >$. If k disjoint matrix products of the size $< m,n,p >$ are computed (sharing no variables), we write $L \to k < m,n,p >$.

In this part, we study the problems of matrix multiplication of the form $< n^r, n^s, n^t >$ with positive integers n and non-negative rational numbers r, s, and t. Let $O(n^{\omega(r,s,t)})$ denote the bilinear complexity of $< n^r, n^s, n^t >$, that is, $O(n^{\omega(r,s,t)})$ bilinear multiplications suffice for solving the problem $< n^r, n^s, n^t >$. The exponent $\omega(r,s,t)$ will be called the (matrix multiplication) exponent for $< n^r, n^s, n^t >$. Due to (1.4), we have

$$\omega(r,s,t) = \omega(t,r,s) = \omega(s,t,r) = \omega(r,t,s) = \omega(s,r,t) = \omega(t,s,r) \ . \tag{2.3}$$

Therefore, it suffices to estimate any one of the six latter exponents for given n, r, s and t.

The exponents $\omega(r,s,t)$ satisfy the following homogeneity equation:

$$\omega(ar,as,at) = a\omega(r,s,t)$$

since

$$O(n^{\omega(ar,as,at)}) = O((n^a)^{\omega(r,s,t)}) = O(n^{a\omega(r,s,t)}) \ .$$

There is the straightforward information lower bound:

$$\omega(r,s,t) \geq \max\{r+s, s+t, t+r\} \ . \tag{2.4}$$

If $r = s = t$, then $< n^r, n^s, n^t >$ represents the problem $< n^r, n^r, n^r >$ of multiplication of a square matrix by a square matrix. Computing its bilinear complexity is reduced to computing the exponent $\omega(r,r,r) = r \cdot \omega(1,1,1)$, that is, to computing $\omega(1,1,1)$, by homogeneity. Current record upper bound $\omega(1,1,1) = \omega < 2.376$ is due to [CW90].

If two values among r, s and t are equal to each other, say, if $r = s \neq t$, then

$$< n^r, n^s, n^t >$$

represents the problem of multiplication of a square matrix by a rectangular matrix. Computing its bilinear complexity is reduced to computing the exponent

$$\omega(r,r,t) = r \cdot \omega(1,1,t/r),$$

that is, to computing $\omega(1,1,t/r)$, by homogeneity. We recall the upper bound

$$\omega(1,1,t/r) = 2 + o(1) \quad \text{for} \quad t/r \leq 0.294, \quad [\text{Co96}],$$

which matches the lower bound $\omega(1,1,t/r) \geq 2$ of (2.2), up to the term $o(1)$.

If r, s and t are distinct from each other, $< n^r, n^s, n^t >$ represents the problem of multiplication of a rectangular matrix by a rectangular matrix. In addition, $< n^r, n^s, n^r >$ $(s \neq r)$ also represents the problem of multiplication of rectangular matrix by rectangular matrix. In this paper, we will present algorithms for multiplication of matrices of such sizes.

We will need the following basic results.

Theorem 2.1 (Schönhage [Sc81]) *Assume given a field* $\mathbf{F}$, *coefficients* $\alpha_{i,j,h,l}$, $\beta_{j,k,h,l}$, $\gamma_{k,i,h,l}$ *in* $\mathbf{F}(\lambda)$ (*the field of rational functions in a single indeterminate* λ), *and polynomials* f_g *over* $\mathbf{F}$, *such that*

$$\sum_{l=1}^{L} \left(\sum_{i,j,h} \alpha_{i,j,h,l} x_{i,j}^{(h)} \right) \left(\sum_{i,j,h} \beta_{j,k,h,l} y_{j,k}^{(h)} \right) \left(\sum_{i,j,h} \gamma_{k,i,h,l} z_{i,j}^{(h)} \right)$$
$$= \sum_{h} \left(\sum_{i=1}^{m_h} \sum_{j=1}^{n_h} \sum_{k=1}^{p_h} x_{i,j}^{(h)} y_{j,k}^{(h)} z_{k,i}^{(h)} \right) + \sum_{g>0} \lambda^g f_g(x_{i,j}^{(h)}, y_{j,k}^{(h)}, z_{k,i}^{(h)})$$

is an identity in $x_{i,j}^{(h)}$, $y_{j,k}^{(h)}$, $z_{k,i}^{(h)}$, λ. *Then, given* $\epsilon > 0$, *one can construct an algorithm to multiply* $N \times N$ *square matrices in* $O(N^{3\tau+\epsilon})$ *operations, where* τ *satisfies*

$$L = \sum_{h} (m_h n_h p_h)^{\tau} .$$

Theorem 2.1 enables us to estimate $\omega(r,s,t)$ from above as soon as we obtain a bilinear algorithm for a disjoint matrix multiplication, in particular, for k disjoint problems

$$< m, n, p > .$$

Theorem 2.2 (Salem and Spencer [SS42]) *Given $\epsilon > 0$, there exists $M_\epsilon \simeq 2^{c/\epsilon^2}$ such that for all $M > M_\epsilon$, there is a set B of $M' > M^{1-\epsilon}$ distinct integers,*

$$0 < b_1 < b_2 < \cdots < b_{M'} < M/2,$$

with no three terms in an arithmetic progression: for any triple of b_i, b_j, $b_k \in B$, we have

$$b_i + b_j = 2b_k \quad \textit{iff} \quad b_i = b_j = b_k \ .$$

In our presentation, we will closely follow the line of [CW90]. In particular, as in [CW90], we will use theorem 2.2 in order to transform tensor product construction into the form $k < m, n, p >$ for sufficiently large k, m, n and p.

3 Basic Algorithm for $< n, n, n^2 >$

In this and the next sections, we will extensively use the techniques of [CW90] (compare [Pan] and [St86] on some preceding work). We begin with a basic algorithm from [CW90], equation (5), which gives us one of the most effective examples of the trilinear aggregating techniques first introduced in [Pan72] (cf. also [Pan] and [Pan,a]). For a given value of the integer q, we will call this construction D_q.

$$\begin{aligned}
&\textstyle\sum_{i=1}^{q} \lambda^{-2}(x_0^{[0]} + \lambda x_i^{[1]})(y_0^{[0]} + \lambda y_i^{[1]})(z_0^{[0]} + \lambda z_i^{[1]}) \\
&-\lambda^{-3}(x_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} x_i^{[1]})(y_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} y_i^{[1]})(z_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} z_i^{[1]}) \\
&+[\lambda^{-3} - q\lambda^{-2}](x_0^{[0]})(y_0^{[0]})(z_0^{[0]}) \\
&= \textstyle\sum_{i=1}^{q}(x_0^{[0]} y_i^{[1]} z_i^{[1]} + x_i^{[1]} y_0^{[0]} z_i^{[1]} + x_i^{[1]} y_i^{[1]} z_0^{[0]}) + O(\lambda).
\end{aligned} \tag{3.1}$$

The x-variables in (3.1) consist of two blocks: $X^{[0]} = \{x_0^{[0]}\}$ and $X^{[1]} = \{x_1^{[1]}, \cdots, x_q^{[1]}\}$. Similarly, the y-variables consist of blocks $Y^{[0]}$ and $Y^{[1]}$, and the z-variables consist of blocks $Z^{[0]}$ and $Z^{[1]}$.

Our next goal is to estimate the exponent $\omega(1,1,2)$.

Consider the $4N^{th}$ tensor power of (3.1). Each variable $x_i^{[I]}$ in the tensor power is the tensor product of $4N$ variables $x_j^{[J]}$, one from each of $4N$ copies of the original algorithm (3.1). j ranges in $\{0,1,2,\cdots,q\}$. The subscript i is a vector of dimension $4N$ formed by the $4N$ subscripts j. J ranges in $\{0,1\}$. The superscript $[I]$ is a vector of dimension $4N$ having entries in $\{0,1\}$, formed by the $4N$ superscripts $[J]$. Clearly, $[I]$ is uniquely determined by i.

In our tensor power, there are 3^{4N} triples $(X^{[I]},Y^{[J]},Z^{[K]})$; each of them is a matrix product of some size $< m,n,p >$ with $mnp = q^{4N}$. We will eliminate some triples by setting to zero some blocks of variables x, y and/or z, so as to stay with some triples of the form $< q^N,q^N,q^{2N} >$ sharing no variables. Then we will estimate the number of the remaining triples, which will define the exponent $\omega(1,1,2)$. When we zero a block $X^{[I]}$ (respectively, $Y^{[I]}$, $Z^{[I]}$), we will set to zero all the x-(respectively, y-, z-) variables with the given superscript pattern.

Hereafter, $\binom{Q}{Q_1,Q_2,\cdots,Q_s}$, for positive integers Q, Q_1, Q_2, $\cdots$, Q_s satisfying

$$Q_1+Q_2+\cdots+Q_s=Q,$$

denote the multinomial expansion coefficient. Our presentation will closely follow section 6 of [CW90].

For all i and I, set $x_i^{[I]}=0$, unless I consists of $2N$ indices of 0 and exactly as many indices of 1. For all j and J, set $y_j^{[J]}=0$ unless J consists of N indices of 0 and $3N$ indices of 1, and similarly for $z_k^{[K]}$. When we complete this procedure, there still remain $\binom{4N}{2N,N,N}$ blocks of triples $(X^{[I]},Y^{[J]},Z^{[K]})$. The blocks are compatible, which means that the locations of their zero indices are disjoint, i.e. among the superscript vectors of $X^{[I]}Y^{[J]}Z^{[K]}$, there is one and only one zero in the location of the same component. (For

example, for $N = 2$, the block $X^{[10110100]}Y^{[11011011]}Z^{[01101111]}$ is compatible). Among them, for each block of variables $Z^{[K]}$, there are $\binom{3N}{2N,N}$ pairs $(X^{[I]}, Y^{[J]})$ sharing this block; for each block $Y^{[K]}$, there are also $\binom{3N}{2N,N}$ pairs $(X^{[I]}, Z^{[K]})$ sharing it; and for each block $X^{[I]}$, there are $\binom{2N}{N,N}$ pairs $(Y^{[J]}, Z^{[K]})$ sharing it. Set $M = 2\binom{3N}{2N,N} + 1$. Select a sufficiently small positive ϵ and a sufficiently large N, so that the latter value M would satisfy the assumptions of the Salem-Spencer theorem for this ϵ; construct a Salem-Spencer set B (cf. [SS42], [Be46], and [CW90]), where the cardinality of B is $M' \geq M^{1-\epsilon}$. In the next section, by revisiting the techniques of section 6 of [CW90], we obtain at least

$$H = \frac{1}{4}\frac{M'}{M^2}\binom{4N}{2N,N,N} \tag{3.2}$$

non-zero block products represented by the triples $(X^{[I]}Y^{[J]}Z^{[K]})$ and pairwise sharing no variables $X^{[I]}$, $Y^{[J]}$ or $Z^{[K]}$.

The fine structure of each block scalar product represents a matrix product of the size

$$< q^N, q^N, (q^N)^2 > \ .$$

For $q^N = n$, this turns into $< n, n, n^2 >$. For example, for $N = 1$, the fine structure of the compatible triple $X^{[1010]}Y^{[1101]}Z^{[0111]}$ is

$$X^{[1010]}_{i0k0}Y^{[1101]}_{ij0l}Z^{[0111]}_{0jkl}\ , \quad i,j,k,l = 1,2,\cdots,q,$$

which represents the matrix product

$$\begin{pmatrix} x_{1010} & \cdots & x_{q010} \\ \vdots & \vdots & \vdots \\ x_{10q0} & \cdots & x_{q0q0} \end{pmatrix} \begin{pmatrix} y_{1101} & \cdots & y_{1q01} & | & \cdots & | & y_{110q} & \cdots & y_{1q0q} \\ \vdots & \vdots & \vdots & | & \cdots & | & \vdots & \vdots & \vdots \\ y_{q101} & \cdots & y_{qq01} & | & \cdots & | & y_{q10q} & \cdots & y_{qq0q} \end{pmatrix} \begin{pmatrix} z_{0111} & \cdots & z_{01q1} \\ \vdots & \vdots & \vdots \\ z_{0q11} & \cdots & z_{0qq1} \\ \vdots & \vdots & \vdots \\ z_{011q} & \cdots & z_{01qq} \\ \vdots & \vdots & \vdots \\ z_{0q1q} & \cdots & z_{0qqq} \end{pmatrix} .$$

We deduce from the above algorithm and from theorem 2.2 that

$$(q+2)^{4N} \geq cHn^{\omega(1,1,2)} , \tag{3.3}$$

where c is the overhead constant of $O(n^{\omega(1,1,2)})$ and H is defined by (3.2). By applying Stirling's formula

$$\lim_{n\to\infty} \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{n!} = 1 \tag{3.4}$$

in order to estimate H, we obtain

$$(q+2)^{4N} \geq c' N^{-\frac{1}{2}(1-\epsilon)} \left(\frac{4^4}{3^3}\right)^N \left(\frac{2^2}{3^3}\right)^{N\epsilon} q^{N\omega(1,1,2)} , \tag{3.5}$$

where c' is a constant. Let $\epsilon \to 0$, $N \to \infty$, take the N^{th} roots and then logarithms of both sides of (3.5), and obtain that

$$(q+2)^4 \geq \left(\frac{4^4}{3^3}\right) q^{\omega(1,1,2)} ,$$

$$\omega(1,1,2) \leq \frac{1}{\log q} \log\left(\frac{27(q+2)^4}{256}\right) .$$

The right-hand side is minimized for $q = 10$:

$$\omega(1,1,2) \leq 3.339848783\cdots \leq 3.3399 . \tag{3.6}$$

4 The Number of Disjoint Nonscalar Block Products

In this section, we will proceed again along the line of section 6 of [CW90] modified slightly so as to estimate $\omega(1,1,2)$, rather than $\omega(1,1,1)$.

Choose integers w_j at random in the interval from 0 to $M-1$, for $j = 0,1,2,\cdots,4N$, and compute the integers

$$b_X(I) \equiv \sum_{j=1}^{4N} I_j w_j \ (\bmod M),$$

$$b_Y(J) \equiv w_0 + \sum_{j=1}^{4N} J_j w_j \ (\bmod M),$$

$$b_Z(K) \equiv (w_0 + \sum_{j=1}^{4N} (2 - K_j) w_j)/2 \ (\bmod M),$$

where $I = (I_1, \cdots, I_{4N}) \in \{0,1\}^{4N}$, I_j is 0 or 1, $j = 1, \cdots, 4N$. As in [CW90], obtain that

$$b_X(I) + b_Y(J) - 2b_Z(K) \equiv 0 \bmod M,$$

for any triple of blocks ($X^{[I]}$, $Y^{[J]}$, $Z^{[K]}$) whose product $X^{[I]}Y^{[J]}Z^{[K]}$ appears in the trilinear form. [Indeed, examine the contribution of each w_j and observe that for each of the three terms

$$x_0^{[0]} y_i^{[1]} z_i^{[1]}, \qquad x_i^{[1]} y_0^{[0]} z_i^{[1]}, \qquad x_i^{[1]} y_i^{[1]} z_0^{[0]},$$

we have $I_j + J_j + K_j = 2$ in the basic construction.]

Set $X^{[I]} = 0$ unless $b_X(I)$ is in the Salem-Spencer set B, set $Y^{[J]} = 0$ unless $b_Y(J) \in B$, and set $Z^{[K]} = 0$ unless $b_Z(K) \in B$. Then, for each triple (I, J, K), where $X^{[I]}Y^{[J]}Z^{[K]} \neq 0$, we have

$$b_X(I) + b_Y(J) \equiv 2b_Z(K) \bmod M, \quad b_X(I),\ b_Y(J),\ b_Z(K) \in B,$$

and therefore,

$$b_X(I) = b_Y(J) = b_Z(K),$$

by the virtue of Salem-Spencer's theorem.

We recall that the block $X^{[I]}$ is the set of q^{4N} variables $x_i^{[I]}$, with nonzero indices in $2N$ specified places, that is, sharing a common superscript I, a nonzero block is one which has not yet been set to zero; blocks $X^{[I]}$, $Y^{[J]}$, $Z^{[K]}$ are compatible if the locations of their zero indices are pairwise disjoint. Let us complete the pruning procedure, as in [CW90]. Make lists of triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$ representing compatible nonzero blocks, with $b_X(I) = b_Y(J) = b_Z(K) = b$ for all $b \in B$. If any triple $(X^{[I]}, Y^{[J]}, Z^{[K]})$ on the list shares a block (say, $Z^{[K]}$) with another triple $(X^{[I']}, Y^{[J']}, Z^{[K']})$ occurring earlier in the list, then eliminate the former triple by setting to zero one of the other blocks (say, $X^{[I]}$). Now, we apply the counting argument of [CW90] and extend the lemma of section 6 of [CW90] as follows:

Lemma 4.1 *The expected number of triples remaining on each list, after pruning, is at least*

$$\frac{1}{4M^2}\binom{4N}{2N,N,N}.$$

Proof: Compare the expected number, $\binom{4N}{2N,N,N}M^{-2}$, of triples in the list before pruning, for each $b \in B$, with the upper estimate

$$\frac{3}{2}\binom{4N}{2N,N,N}\left(\binom{2N}{N,N}-1\right)M^{-3}$$

for the expected number of unordered pairs of compatible triples sharing a Z-block, a Y-block, or an X-block. The latter number is an upper bound on the expected number of eliminated pairs of triples, which is easily showed to be not less than the expected number of eliminated triples. Comparison of the two upper estimates gives us Lemma 4.1. □

It follows from Lemma 4.1 that the expected number of triples remaining on all lists after pruning (average over all the choices of w_j) is at least H of (3.2). Therefore, we may fix a choice of w_j that achieves at least as many triples on the list.

The procedure of computing H can be summarized in the following way:

Procedure 4.1

Step 1: First compute the number of triples of blocks, having a fixed patern $< n^r, n^s, n^t >$ among all the triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$ that we have after taking the tensor power of a given basic trilinear algorithm [like (3.1)]. In section 3, $< n^r, n^s, n^t >=< n, n, n^2 >$, and there are $\binom{4N}{2N, N, N}$ special triples among a total of 3^{4N}.

Step 2: Compute the numbers of pairs $(X^{[I]}, Y^{[J]})$ sharing a single block $Z^{[K]}$, of pairs $(X^{[I]}, Z^{[K]})$ sharing a single block $Y^{[J]}$, and of pairs $(Y^{[J]}, Z^{[K]})$ sharing a single block $X^{[I]}$ (in section 3, these numbers are

$$\binom{3N}{2N, N}, \quad \binom{3N}{2N, N}, \quad \binom{2N}{N, N},$$

respectively). Determine the largest of them (here, the largest is $\binom{3N}{2N, N}$).

Step 3: Perform the pruning procedure extending the one presented in this section in the straightforward way and show that there still remain at least

$$H = \frac{\text{the number from step 1}}{4 \times \text{the largest from step 2}}$$

triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$ sharing no variables.

The latter procedure will be repeatedly applied in the next sections.

5 Improved Algorithm for $< n, n, n^2 >$

In this section, we will improve our upper bound on the exponent $\omega(1,1,2)$ from 3.3399 to 3.333953 by combining the technique of Section 7 of [CW90] and the same ideas as in the previous section. The improvement will be due to using a more complicated starting algorithm, that is, the basic algorithm from [CW90], equation (10):

$$\begin{aligned}
&\textstyle\sum_{i=1}^{q} \lambda^{-2}(x_0^{[0]} + \lambda x_i^{[1]})(y_0^{[0]} + \lambda y_i^{[1]})(z_0^{[0]} + \lambda z_i^{[1]}) \\
&-\lambda^{-3}(x_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} x_i^{[1]})(y_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} y_i^{[1]})(z_0^{[0]} + \lambda^2 \textstyle\sum_{i=1}^{q} z_i^{[1]}) \\
&+[\lambda^{-3} - q\lambda^{-2}](x_0^{[0]} + \lambda^3 x_{q+1}^{[2]})(y_0^{[0]} + \lambda^3 y_{q+1}^{[2]})(z_0^{[0]} + \lambda^3 z_{q+1}^{[2]}) \qquad (5.1) \\
&= \textstyle\sum_{i=1}^{q} (x_0^{[0]} y_i^{[1]} z_i^{[1]} + x_i^{[1]} y_0^{[0]} z_i^{[1]} + x_i^{[1]} y_i^{[1]} z_0^{[0]}) \\
&+x_0^{[0]} y_0^{[0]} z_{q+1}^{[2]} + x_0^{[0]} y_{q+1}^{[2]} z_0^{[0]} + x_{q+1}^{[2]} y_0^{[0]} z_0^{[0]} + O(\lambda).
\end{aligned}$$

The subscripts now form three classes: $\{0\}$, $\{q+1\}$ and $\{1, 2, \cdots, q\}$, which will again be denoted i. Again, the subscripts uniquely determine the superscripts (block indices).

Take the $4N^{th}$ power of this construction. Each variable $x_i^{[I]}$ in the tensor power is the tensor product of $4N$ variables $x_j^{[J]}$, one from each of $4N$ copies of the original algorithm (5.1). Its subscript i is a vector of dimension $4N$ with entries in $\{0, 1, 2, \cdots, q, q+1\}$, formed by the $4N$ subscripts j. Its superscript $[I]$ is a vector of dimension $4N$ with entries in $\{0, 1, 2\}$, formed by the $4N$ superscripts $[J]$.

Set $L = \lceil \beta N \rceil$, where β is a small positive number (which will be specified later on, roughly at the level of 0.02). As in the previous section, we currently have 6^{4N} triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. Set $x_i^{[I]} = 0$, unless I has exactly $2N$ indices of 0, exactly $2N - 2L$ indices of 1, and exactly $2L$ indices of 2; set $y_j^{[J]} = 0$, unless J has exactly $N + 2L$ indices of 0, exactly $3N - 3L$ indices of 1, and exactly L indices of 2, and similarly for $z_k^{[K]}$. When we

complete this procedure, there still remain

$$\binom{4N}{L, L, 2L, 2N-2L, N-L, N-L}$$

blocks of triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. Namely, among the $4N$ copies of construction (5.1), we pick

$x_0^{[0]} y_i^{[1]} z_i^{[1]}$ from $2N-2L$ copies,

$x_i^{[1]} y_0^{[0]} z_i^{[1]}$ from $N-L$ copies,

$x_i^{[1]} y_i^{[1]} z_0^{[0]}$) from $N-L$ copies,

$x_0^{[0]} y_0^{[0]} z_{q+1}^{[2]}$ from L copies,

$x_0^{[0]} y_{q+1}^{[2]} z_0^{[0]}$ from L copies and

$x_{q+1}^{[2]} y_0^{[0]} z_0^{[0]}$ from $2L$ copies.

They are compatible, which means that the sum of indices at the same locations of their superscripts I, J and K is 2. Among them, for each $Z^{[K]}$, there are

$$\binom{3N-3L}{2N-2L, N-L}\binom{N+2L}{N-L, 2L, L}$$

pairs $(X^{[I]}, Y^{[J]})$ sharing it; for each $Y^{[K]}$, there are as many pairs $(X^{[I]}, Z^{[K]})$ sharing it; but for each $X^{[I]}$, there are only

$$\binom{2N}{2N-2L, L, L}\binom{2N-2L}{N-L, N-L}$$

pairs $(Y^{[J]}, Z^{[K]})$ sharing it.

Select the larger (that is, the former) of the two numbers of pairs and set

$$M = 2\binom{3N-3L}{2N-2L, N-L}\binom{N+2L}{N-L, 2L, L} + 1 \ .$$

Construct a Salem-Spencer set B. Select random integers $0 \le w_j < M$, $j = 0, 1, 2, \cdots, 4N$. Then, by following the lines of section 7 of [CW90] and of our section 4, in particular, by

applying Procedure 4.1, we obtain at least

$$H^* = \frac{1}{4}\frac{M'}{M^2}\begin{pmatrix} 4N \\ L, L, 2L, 2N-2L, N-L, N-L \end{pmatrix}$$

non-zero triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$, which share no variables with each other, where $M' \geq M^{1-\epsilon}$, for a fixed positive ϵ, is the cardinality of B. Each of these triples corresponds to a matrix product of size

$$< q^{N-L}, q^{N-L}, (q^{N-L})^2 > \ ,$$

which turns into $< n, n, n^2 >$ for $n = q^{N-L}$. Letting $M(n,n,n^2) = O(n^{\omega(1,1,2)})$ and summarizing our estimates, we obtain

$$(q+2)^{4N} \geq cH^* q^{(N-L)\omega(1,1,2)} \ .$$

Applying Stirling's formula to the value H^*, we obtain that

$$(q+2)^{4N} \geq cN^{-1+\frac{3}{2}\epsilon}\left[\frac{256}{\beta^\beta(3-3\beta)^{(3-3\beta)}(1+2\beta)^{(1+2\beta)}}\right]^N (c')^N q^{N(1-\beta)\omega(1,1,2)} \ .$$

Let $\epsilon \to 0$, $N \to \infty$, take N^{th} roots and then logarithms on both sides and deduce that

$$(q+2)^4 \geq \frac{256}{\beta^\beta(3-3\beta)^{(3-3\beta)}(1+2\beta)^{(1+2\beta)}} q^{(1-\beta)\omega(1,1,2)} \ ,$$

$$\omega(1,1,2) \leq \frac{1}{(1-\beta)\log q}\log\left(\frac{\beta^\beta(3-3\beta)^{(3-3\beta)}(1+2\beta)^{(1+2\beta)}(q+2)^4}{256}\right) \ .$$

$q = 9$ and $\beta = 0.016$ minimize the right-hand side of the latter inequality, and we obtain that

$$\omega(1,1,2) \leq 3.333953\cdots < 3.334 \ .$$

6 Basic Algorithm for $< n^r, n^s, n^t >$

In this section, we will combine the ideas and techniques of sections 3 and 4 so as to develop the basic algorithms for estimating the exponents of rectangular matrix multipli-

cations of arbitrary shape, that is, for the problem $< n^r, n^s, n^t >$. For convenience, we first classify the triples $< n^r, n^s, n^t >$, for all rational r, s, t as follows:

(1) $< n^r, n, n >$ with $r > 1$;

(2) $< n, n, n^t >$ with $0 \leq t \leq 1$;

(3) $< n^r, n, n^t >$ with $r > 1 > t > 0$.

Indeed, we have three respective classes of triples:

(1) Among r, s, t, two are equal and the third one is larger. In this case, we may assume $r > s = t$ [cf. (2.1)]. Then, by homogeneity of the exponent, $\omega(r, s, t) = s\omega(r/s, 1, 1)$, $r/s > 1$.

(2) Among r, s, t, two are equal and the third one is not larger. In this case, we may assume $r = s \geq t$. Then, by homogeneity of the exponent, $\omega(r, s, t) = r\omega(1, 1, t/r)$, $0 \leq t/r \leq 1$.

(3) Among r, s, t, all three are pairwise distinct. In this case, we may assume $r > s > t$. Then, by homogeneity of the exponent, $\omega(r, s, t) = s\omega(r/s, 1, t/s)$, $r/s > 1 > t/s > 0$.

6.1 The case $< n^r, n, n >$ with $r > 1$

Due to (1.4), we may assume that $< n, n, n^r >$ is case (1). We begin with the construction (3.1) again. Take the $(2+r)N^{th}$ tensor power of (3.1), where N is sufficiently large so that $(2+r)N$ is an integer. Each variable $x_i^{[I]}$ in the tensor power is the tensor product of $(2+r)N$ variables $x_j^{[J]}$, one from each of $(2+r)N$ copies of the original algorithm (3.1). Its subscript i is a vector of dimension $(2+r)N$ with entries in $\{0, 1, 2, \cdots, q\}$, made up of the $(2+r)N$ subscripts j. Its superscript $[I]$ is a vector of dimension $(2+r)N$ with entries in $\{0, 1\}$, made up of the $(2+r)N$ superscripts $[J]$. Clearly, $[I]$ is uniquely determined by i.

In our tensor power, there are totally $3^{N(2+r)}$ triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. We will eliminate

some triples and preserve those of dimension $< q^N, q^N, (q^N)^r >$, sharing no variables with each other. Then we will estimate the number of the remaining triples.

Set $x_i^{[I]} = 0$ unless I has exactly rN indices of 0 and exactly $2N$ indices of 1, set $y_j^{[J]} = 0$ unless J has exactly N indices of 0 and exactly $(1+r)N$ indices of 1, and similarly for $z_k^{[K]}$. When we complete this procedure, there still remain $\binom{(2+r)N}{N, N, rN}$ blocks of triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. They are compatible, which means that the locations of their zero indices are disjoint. Among them, for each $Z^{[K]}$, there are $\binom{(1+r)N}{N, rN}$ pairs $(X^{[I]}, Y^{[J]})$ sharing it; for each $Y^{[K]}$, there are as many pairs $(X^{[I]}, Z^{[K]})$ sharing it; for each $X^{[I]}$, there are only $\binom{2N}{N, N}$ pairs $(Y^{[J]}, Z^{[K]})$ sharing it. We select the larger (former) of the two latter estimates and set

$$M = 2\binom{(1+r)N}{N, rN} + 1 .$$

Construct a Salem-Spencer set B (cf. [SS42] and [Be46]), where the cardinality of B is $M' \geq M^{1-\epsilon}$. In the same way as in the previous sections, we obtain at least

$$\tilde{H} = \frac{1}{4}\frac{M'}{M^2}\binom{(2+r)N}{N, N, rN}$$

non-zero triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$ sharing no variables with each other, that is, our algorithm computes at least $\tilde{H}$ block products $(X^{[I]}, Y^{[J]}, Z^{[K]})$. The fine structure of each block product is a matrix product of size

$$< q^N, q^N, (q^N)^r > ,$$

which is $< n, n, n^r >$ for $q^N = n$. It follows that

$$(q+2)^{(2+r)N} \geq c\tilde{H}n^{\omega(1,1,r)} ,$$

where c is the overhead constant of $O(n^{\omega(1,1,r)})$. Applying Stirling's formula to approximate

$\tilde{H}$, we obtain

$$(q+2)^{(2+r)N} \geq cN^{-\frac{1}{2}(1-\epsilon)} \left(\frac{(2+r)^{(2+r)}}{(1+r)^{(1+r)}} \right)^N (c')^{N\epsilon} q^{N\omega(1,1,r)} ,$$

where c and c' are constants. Let $\epsilon \to 0$, $N \to \infty$, take N^{th} roots, and obtain

$$(q+2)^{(2+r)} \geq \left(\frac{(2+r)^{(2+r)}}{(1+r)^{(1+r)}} \right) q^{\omega(1,1,r)} .$$

By solving for $\omega(1,1,r)$, we obtain

$$\omega(1,1,r) \leq \frac{1}{\log q} \log \left(\frac{(1+r)^{(1+r)}(q+2)^{(2+r)}}{(2+r)^{(2+r)}} \right) . \tag{6.1}$$

6.2 The Case $< n, n, n^t >$ with $0 \leq t \leq 1$

We replace t by r, for convenience. In this case the algorithm is almost completely the same as in the case $r > 1$. The small difference is that we now set

$$M = 2 \begin{pmatrix} 2N \\ N, N \end{pmatrix} + 1 ,$$

since $\begin{pmatrix} 2N \\ N, N \end{pmatrix}$ exceeds $\begin{pmatrix} (1+r)N \\ N, rN \end{pmatrix}$. We proceed as in subsection 6.1 and obtain that

$$\omega(1,1,r) \leq \frac{1}{\log q} \log \left(\frac{2^2 r^r (q+2)^{(2+r)}}{(2+r)^{(2+r)}} \right) , \tag{6.2}$$

for $0 \leq r \leq 1$.

6.3 The Case $< n^r, n, n^t >$ with $r > 1 > t > 0$

Due to (1.4), we may assume that $< n^t, n, n^r >$ with $r > 1 > t > 0$, instead of $< n^r, n, n^t >$ with $r > 1 > t > 0$. In this case, we take the $(t+1+r)N^{th}$ tensor power of (3.1), where N is sufficiently large so that $(t+1+r)N$ is an integer. In our tensor power, there are a total of $3^{N(t+1+r)}$ triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. As before, we will eliminate some

triples and preserve those of the dimension $< (q^N)^t, q^N, (q^N)^r >$ sharing no variables with each other. Then we will estimate the number of the remaining triples.

Set $x_i^{[I]} = 0$ unless I has exactly rN indices of 0 and exactly $(t+1)N$ indices of 1, set $y_j^{[J]} = 0$ unless J has exactly tN indices of 0 and exactly $(1+r)N$ indices of 1, and set $z_k^{[K]} = 0$ unless K has exactly N indices of 0 and exactly $(t+r)N$ indices of 1. When we complete this procedure, there still remain $\binom{(t+1+r)N}{tN, N, rN}$ blocks of triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. They are compatible, which means that the locations of their zero indices are disjoint. Among them, for each $Z^{[K]}$, there are $\binom{(t+r)N}{tN, rN}$ pairs $(X^{[I]}, Y^{[J]})$ sharing it; for each $Y^{[J]}$, there are $\binom{(1+r)N}{N, rN}$ pairs $(X^{[I]}, Z^{[K]})$ sharing it; for each $X^{[I]}$, there are $\binom{(t+1)N}{tN, N}$ pairs $(Y^{[J]}, Z^{[K]})$ sharing it.

Since $r > 1 > t > 0$, the second of these three estimates is the largest. So we set

$$M = 2\binom{(1+r)N}{N, rN} + 1 .$$

Similarly to subsection 6.1, we obtain that

$$\omega(t, 1, r) \leq \frac{1}{\log q} \log \left(\frac{(1+r)^{(1+r)} t^t (q+2)^{(t+1+r)}}{(t+1+r)^{(t+1+r)}} \right) . \tag{6.3}$$

7 Improved Algorithm for $< n^r, n^s, n^t >$

In this section, we will improve our algorithm of section 6 for the problem $< n^r, n^s, n^t >$ by combining the ideas from sections 5 and 6. We break this section into three subsections and respectively discuss the three cases, as in section 6.

7.1 The case $< n, n, n^r >$ with $r > 1$

We begin with the construction (5.1). Take the $(2+r)N^{th}$ tensor power of this construction, where N is sufficiently large so that $(2+r)N$ is an integer. Each variable $x_i^{[I]}$ in the tensor power is the tensor product of $(2+r)N$ variables $x_j^{[J]}$, one from each of $(2+r)N$ copies of the original algorithm (5.1). The subscript i is a vector of dimension $(2+r)N$ with entries in $\{0, 1, 2, \cdots, q, q+1\}$, made up of the $(2+r)N$ subscripts j. The superscript $[I]$ is a vector of dimension $(2+r)N$ with entries in $\{0, 1, 2\}$, consisting of the $(2+r)N$ superscripts $[J]$.

Set $L = \lceil \beta N \rceil$, where β is a small number to be determined later on (roughly at the level between 0.005 and 0.05). We currently have $6^{(2+r)N}$ triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$. Set $x_i^{[I]} = 0$ unless I has exactly $r(N-L)+2L$ indices of 0, exactly $2(N-L)$ indices of 1 and exactly rL indices of 2; set $y_j^{[J]} = 0$ unless J has exactly $N+rL$ indices of 0, exactly $(1+r)(N-L)$ indices of 1 and exactly L indices of 2, and similarly for $z_k^{[K]}$. When this procedure is completed, there still remain

$$\binom{(2+r)N}{L, L, rL, r(N-L), (N-L), (N-L)}$$

blocks of triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$, which means that, among the $(2+r)N$ copies of construction (5.1), we pick

$x_0^{[0]} y_i^{[1]} z_i^{[1]}$ from $r(N-L)$ copies,

$x_i^{[1]} y_0^{[0]} z_i^{[1]}$ from $(N-L)$ copies,

$x_i^{[1]} y_i^{[1]} z_0^{[0]}$ from $(N-L)$ copies,

$x_0^{[0]} y_0^{[0]} z_{q+1}^{[2]}$ from L copies,

$x_0^{[0]} y_{q+1}^{[2]} z_0^{[0]}$ from L copies, and

$x_{q+1}^{[2]} y_0^{[0]} z_0^{[0]}$ from rL copies.

They are compatible, which means that the sum of indices at the same locations of their superscripts I, J and K is 2. Among them, for each $Z^{[K]}$, there are

$$\begin{pmatrix} (1+r)(N-L) \\ (N-L), r(N-L) \end{pmatrix} \begin{pmatrix} N+rL \\ (N-L), L, rL \end{pmatrix}$$

pairs $(X^{[I]}, Y^{[J]})$ sharing it; for each $Y^{[K]}$, there are as many pairs $(X^{[I]}, Z^{[K]})$ sharing it; for each $X^{[I]}$, there are only

$$\begin{pmatrix} r(N-L)+2L \\ r(N-L), L, L \end{pmatrix} \begin{pmatrix} 2(N-L) \\ (N-L), (N-L) \end{pmatrix}$$

pairs $(Y^{[J]}, Z^{[K]})$ sharing it.

We select the larger former bound and set

$$M = 2 \begin{pmatrix} (1+r)(N-L) \\ (N-L), r(N-L) \end{pmatrix} \begin{pmatrix} N+rL \\ (N-L), L, rL \end{pmatrix} + 1 .$$

Construct a Salem-Spencer set B. Select random integers

$$0 \leq w_j < M, \quad j = 0, 1, 2, \cdots, (2+r)N.$$

As before, we obtain at least

$$\hat{H} = \frac{1}{4} \frac{M'}{M^2} \begin{pmatrix} (2+r)N \\ L, L, rL, r(N-L), (N-L), (N-L) \end{pmatrix}$$

non-zero triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$, which share no variables with each other, where M' is the cardinality of B and $M' \geq M^{1-\epsilon}$. Each of them corresponds to a matrix product of size

$$< q^{(N-L)}, q^{(N-L)}, q^{r(N-L)} > .$$

For $n = q^{(N-L)}$, this turns into $< n, n, n^r >$. Letting $M(n, n, n^r) = O(n^{\omega(1,1,r)})$ and summarizing, we obtain

$$(q+2)^{(2+r)N} \geq c\hat{H} q^{(N-L)\omega(1,1,r)} .$$

Applying Stirling's formula to approximate the value of right-hand side, we have

$$(q+2)^{(2+r)N} \geq cN^{-1+\frac{3}{2}\epsilon}\left[\frac{(2+r)^{(2+r)}}{\beta^{\beta}((1+r)(1-\beta))^{(1+r)(1-\beta)}(1+r\beta)^{(1+r\beta)}}\right]^{N}(c')^{\epsilon N}q^{N(1-\beta)\omega(1,1,r)}.$$

Letting $\epsilon \to 0, N \to \infty$, and taking N^{th} roots, we obtain

$$(q+2)^{(2+r)} \geq \frac{(2+r)^{(2+r)}}{\beta^{\beta}((1+r)(1-\beta))^{(1+r)(1-\beta)}(1+r\beta)^{(1+r\beta)}}q^{(1-\beta)\omega(1,1,r)}.$$

Taking logarithms on both sides and solving for $\omega(1,1,r)$, we obtain the estimate

$$\omega(1,1,r) \leq \frac{1}{(1-\beta)\log q}\log\left(\frac{\beta^{\beta}((1+r)(1-\beta))^{(1+r)(1-\beta)}(1+r\beta)^{(1+r\beta)}(q+2)^{(2+r)}}{(2+r)^{(2+r)}}\right). \tag{7.1}$$

7.2 The Case $< n, n, n^r >$ with $0 \leq r \leq 1$

We treat this case similarly to the case $r > 1$. The small difference is that now

$$\binom{(1+r)(N-L)}{(N-L),r(N-L)}\binom{N+rL}{(N-L),L,rL} < \binom{r(N-L)+2L}{r(N-L),L,L}\binom{2(N-L)}{(N-L),(N-L)}.$$

Therefore, we set

$$M = 2\binom{r(N-L)+2L}{r(N-L),L,L}\binom{2(N-L)}{(N-L),(N-L)}+1.$$

In the same way as in the preceeding subsection, we obtain the exponent bound

$$\omega(1,1,r) \leq \frac{1}{(1-\beta)\log q}\log\left(\frac{(r\beta)^{(r\beta)}(2(1-\beta))^{2(1-\beta)}(r(1-\beta)+2\beta)^{(r(1-\beta)+2\beta)}(q+2)^{(2+r)}}{(2+r)^{(2+r)}}\right). \tag{7.2}$$

7.3 The Case $< n^r, n, n^t >$ with $r > 1 > t > 0$

Due to (1.4), we will discuss $< n^t, n, n^r >$ with $r > 1 > t > 0$, instead of $< n^r, n, n^t >$ with $r > 1 > t > 0$. In this case, take the $(t+1+r)N^{th}$ tensor power of (5.1), where N is sufficiently large, so that $(t+1+r)N$ is an integer. Each variable $x_i^{[I]}$ in the tensor power

is the tensor product of $(t+1+r)N$ variables $x_j^{[J]}$, one from each of $(t+1+r)N$ copies of the original algorithm (5.1). The subscript i is a vector of dimension $(t+1+r)N$ with entries in $\{0,1,2,\cdots,q,q+1\}$, made up of the $(t+1+r)N$ subscripts j. The superscript $[I]$ is a vector of dimension $(t+1+r)N$ with entries in $\{0,1,2\}$, made up of the $(t+1+r)N$ superscripts $[J]$.

Set $L=\lceil \beta N\rceil$, where a small number β will be determined later on (roughly at the level between 0.005 and 0.05). We currently have $6^{(t+1+r)N}$ triples $(X^{[I]},Y^{[J]},Z^{[K]})$. Set $x_i^{[I]}=0$ unless I has exactly $tL+L+r(N-L)$ indices of 0, exactly $(t+1)(N-L)$ indices of 1 and exactly rL indices of 2; set $y_j^{[J]}=0$ unless J has exactly $t(N-L)+L+rL$ indices of 0, exactly $(1+r)(N-L)$ indices of 1, and exactly tL indices of 2; set $z_k^{[K]}=0$ unless K has exactly $tL+(N-L)+rL$ indices of 0, exactly $(t+r)(N-L)$ indices of 1 and exactly L indices of 2. When we complete this procedure, there still remain at least

$$\binom{(t+1+r)N}{tL,L,rL,t(N-L),(N-L),r(N-L)}$$

blocks of triples $(X^{[I]},Y^{[J]},Z^{[K]})$. In accordance with this estimate, among the $(t+1+r)N$ copies of construction (5.1), we pick

$x_0^{[0]}y_i^{[1]}z_i^{[1]}$ from $r(N-L)$ copies,

$x_i^{[1]}y_0^{[0]}z_i^{[1]}$ from $t(N-L)$ copies,

$x_i^{[1]}y_i^{[1]}z_0^{[0]}$ from $(N-L)$ copies,

$x_0^{[0]}y_0^{[0]}z_{q+1}^{[2]}$ from L copies,

$x_0^{[0]}y_{q+1}^{[2]}z_0^{[0]}$ from tL copies, and

$x_{q+1}^{[2]}y_0^{[0]}z_0^{[0]}$ from rL copies.

They are compatible, which means that the sum of indices at the same locations of their

superscripts I, J and K is 2. Among them, for each block $Z^{[K]}$, there are

$$\binom{(t+r)(N-L)}{t(N-L),r(N-L)}\binom{tL+(N-L)+rL}{tL,(N-L),rL}$$

pairs $(X^{[I]},Y^{[J]})$ sharing it; for each $Y^{[K]}$, there are

$$\binom{(1+r)(N-L)}{(N-L),r(N-L)}\binom{t(N-L)+L+rL}{t(N-L),L,rL}$$

pairs $(X^{[I]},Z^{[K]})$ sharing it; for each $X^{[I]}$, there are

$$\binom{(t+1)(N-L)}{t(N-L),(N-L)}\binom{tL+L+r(N-L)}{tL,L,r(N-L)}$$

pairs $(Y^{[J]},Z^{[K]})$ sharing it.

Since $r>1>t>0$, the largest of these three bounds is the second one. So, we set

$$M=2\binom{(1+r)(N-L)}{(N-L),r(N-L)}\binom{t(N-L)+L+rL}{t(N-L),L,rL}+1\,.$$

Along the line of subsection 7.1, we now obtain the exponent bound

$$\omega(t,1,r)\le\frac{1}{(1-\beta)\log q}\times$$
$$\log\left(\frac{(t\beta)^{t\beta}((1+r)(1-\beta))^{(1+r)(1-\beta)}(t(1-\beta)+(1+r)\beta)^{(t(1-\beta)+(1+r)\beta)}(q+2)^{(t+1+r)}}{(t+1+r)^{(t+1+r)}}\right)\,. \tag{7.3}$$

8 Discussion on Optimization of Algorithms for Fast Rectangular Matrix Multiplications

In this section, we will compare our algorithms for rectangular matrix multiplication of this paper with other possible effective algorithms and will choose some combination of our designs so as to optimize the exponents. We will discuss three cases, as in sections 6 and 7.

8.1 The case $< n, n, n^r >$ with $r > 1$

In this case, if we apply square matrix multiplication algorithm (cf. [CW90]), we obtain

$$M(n,n,n^r) = n^{r-1}M(n,n,n) = n^{r-1}O(n^\omega) = O(n^{r-1+\omega}) .$$

Due to $\omega < 2.376$ ([CW90]), $\omega(1,1,r) = r - 1 + \omega < r + 1.376$. Let $g(r) = r + 1.376$, then $g(r)$ is an increasing linear function in the interval $[1,\infty)$ and passes through the points $(1, 2.376)$ and $(2, 3.376)$, where $g(1) = 2.375477\cdots$ agrees with the result of section 8 of [CW90].

Let $f(r)$ denote the right-hand side of (7.1), that is, the exponent estimate for $< n, n, n^r >$ based on the algorithm of subsection 7.1. By combining the results of section 5 and 7, we obtain that $f(r)$ is an increasing function in the interval $[1, +\infty)$ passing through the points (1, 2.38719) and (2, 3.334). For $r = 1$, $f(1) = 2.38719$ agrees with the result of section 7 of [CW90], and $f(2) = 3.334$ agrees with the result of section 5. Near the point $r = 1.171$, we have $f(r) \approx g(r) = r + 1.376$. For $q = 7$ and $\beta = 0.0336$, $f(1.171) = 2.546462806\cdots < g(1.171) = 2.546477\cdots$.

According to this examination, (7.1) minimizes the exponent for $r \geq 1.171 - \epsilon$ for an appropriate small positive ϵ.

8.2 The Case $< n, n, n^r >$ with $0 \leq r \leq 1$

In this case, we let $f(r)$ be the right-hand side of (7.2). $f(r)$ is a monotone increasing continuous function in the interval [0, 1] passing through the points (0, $2+\epsilon$) and (1, 2.38719). The exponent estimate given by $f(r)$ for $r \in [0,1]$ is not yet the best, however. A better

exponent bound for $r \in [0,1]$ is given by

$$\omega(1,1,r) = \begin{cases} 2+o(1)\,, & 0 \le r \le 0.294 = \alpha, \\ \frac{2(1-r)+(r-\alpha)\omega}{1-\alpha}\,, & 0.294 < r \le 1\,. \end{cases} \tag{8.1}$$

Here is its derivation:

$\omega(1,1,r) \le 2+o(1),\ 0 \le r \le 0.294 = \alpha$ comes from [Co96], and we also have

$$\omega(1,1,r) \le \frac{2(1-r)+(r-\alpha)\omega}{1-\alpha},\quad \alpha = 0.294 < r \le 1\,.$$

Indeed,

$$\begin{aligned} & M(n,n,n^r) \\ &= M(n^{\frac{1-r}{1-\alpha}}\cdot n^{\frac{r-\alpha}{1-\alpha}}, n^{\frac{1-r}{1-\alpha}}\cdot n^{\frac{r-\alpha}{1-\alpha}}, n^{\frac{(1-r)\alpha}{1-\alpha}}\cdot n^{\frac{r-\alpha}{1-\alpha}}) \\ &\le M(n^{\frac{1-r}{1-\alpha}}, n^{\frac{1-r}{1-\alpha}}, n^{\frac{(1-r)\alpha}{1-\alpha}})\cdot M(n^{\frac{r-\alpha}{1-\alpha}}, n^{\frac{r-\alpha}{1-\alpha}}, n^{\frac{r-\alpha}{1-\alpha}}) \\ &= O((n^{\frac{1-r}{1-\alpha}})^{2+\epsilon}(n^{\frac{r-\alpha}{1-\alpha}})^{\omega}) \\ &= O(n^{\frac{2(1-r)+(r-\alpha)\omega}{1-\alpha}})\,. \end{aligned}$$

Summarizing the two cases above, we have the optimal choice of our parameters represented by the curves of **Figure 1.**

8.3 The Case $< n^t, n, n^r >$ with $r > 1 > t > 0$

In this case, we first deduce a small upper bound on the exponent $\omega(t,1,r)$. [For lower bound, see (2.4).]

Theorem 8.1 *Let $\omega(t,1,r)$ be the exponent of $< n^t, n, n^r >$. Then*

$$\omega(t,1,r) = \begin{cases} r+1+\epsilon\,, & 0 \le t \le 0.294 = \alpha\,, \\ \frac{r(1-\alpha)+(1-t)+(\omega-1)(t-\alpha)}{1-\alpha}\,, & 0.294 < t \le 1\,. \end{cases} \tag{8.2}$$

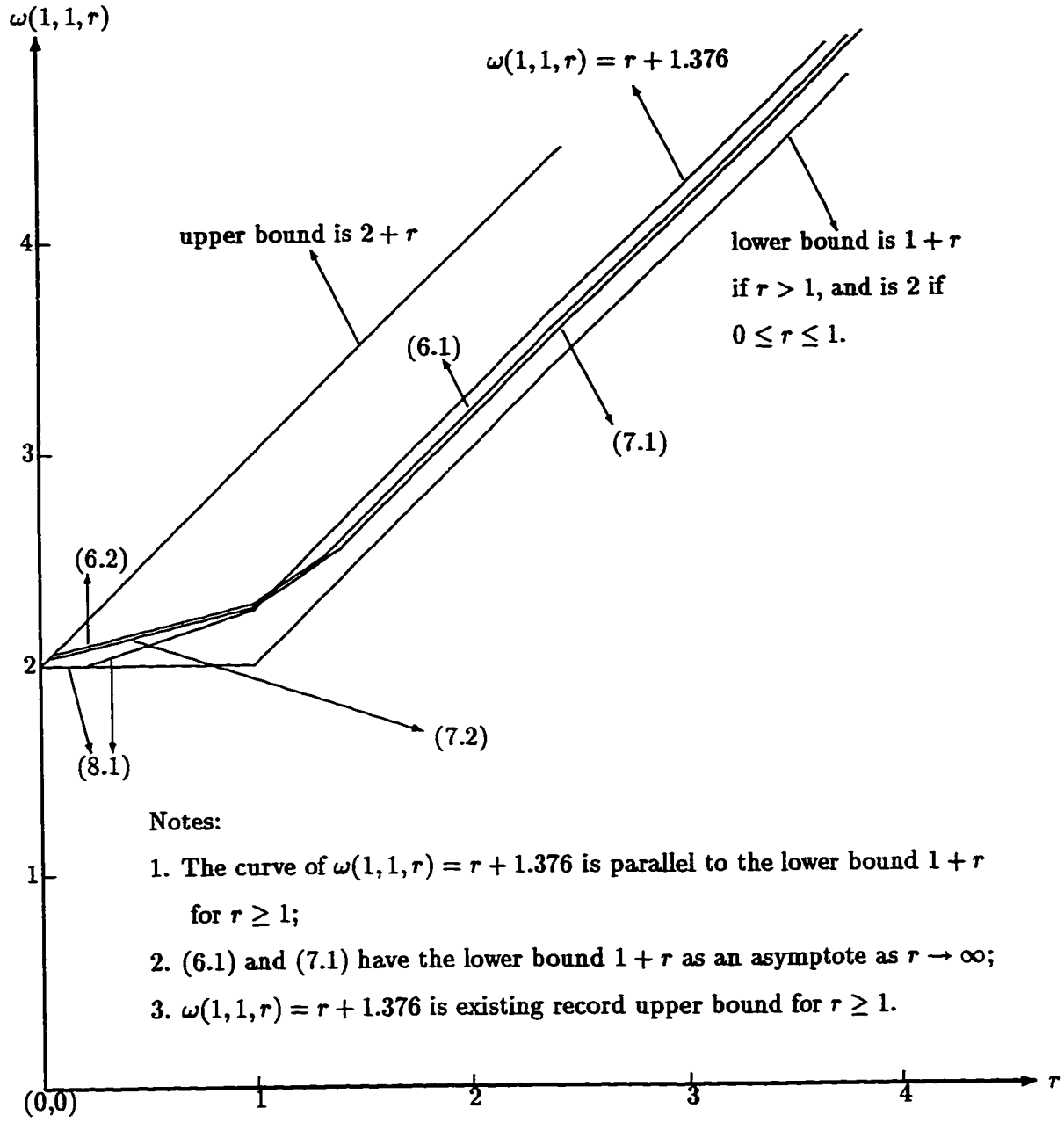

Figure 1: (6.1), (6.2), (7.1), (7.2) and (8.1) refer to the respective equations of this paper.

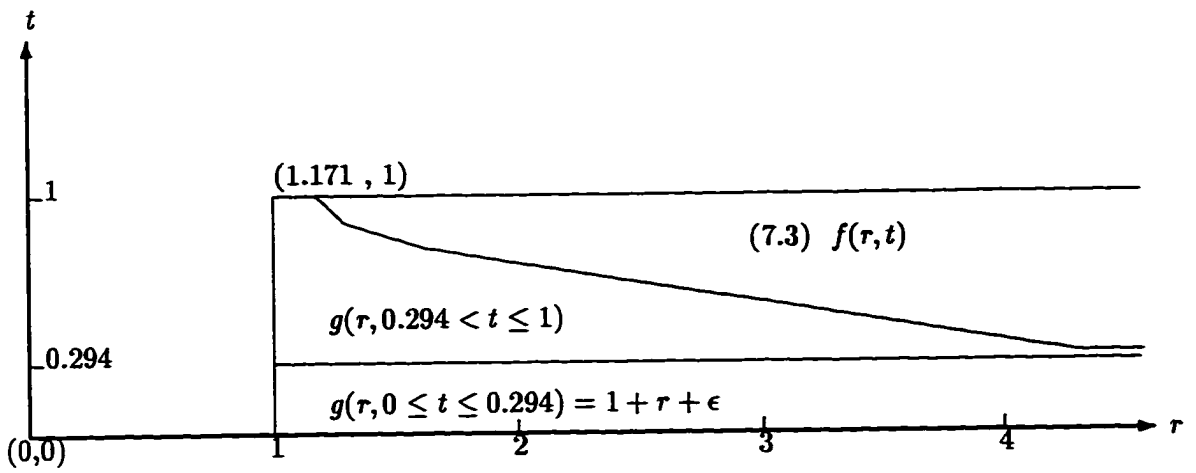

Figure 2: the three areas are, respectively, the optimal region of the three exponent functions for $< n^t, n, n^r >$, $0 \leq t \leq 1 \leq r$.

Proof: For $0 \leq t \leq 0.294 = \alpha$, we have

$$\begin{aligned} & M(n^t, n, n^r) \\ &= n^{r-1} M(n, n, n^t) \\ &\leq n^{r-1} M(n, n, n^\alpha) \\ &= n^{r-1} O(n^{2+\epsilon}) \quad \text{(cf. [Co96])} \\ &= O(n^{r+1+\epsilon}) , \end{aligned}$$

that is, $\omega(t, 1, r) = r + 1 + \epsilon$.

For $\alpha = 0.294 < t \leq 1$, the current best exponent estimate can be derived as follows:

$$\begin{aligned} & M(n^t, n, n^r) = M(n^r, n, n^t) \\ &= M(n^{r - \frac{t-\alpha}{1-\alpha}} \cdot n^{\frac{t-\alpha}{1-\alpha}}, n^{\frac{1-t}{1-\alpha}} \cdot n^{\frac{t-\alpha}{1-\alpha}}, n^{\frac{(1-t)\alpha}{1-\alpha}} \cdot n^{\frac{t-\alpha}{1-\alpha}}) \\ &\leq M(n^{r - \frac{t-\alpha}{1-\alpha}}, n^{\frac{1-t}{1-\alpha}}, n^{\frac{(1-t)\alpha}{1-\alpha}}) \cdot M(n^{\frac{t-\alpha}{1-\alpha}}, n^{\frac{t-\alpha}{1-\alpha}}, n^{\frac{t-\alpha}{1-\alpha}}) \\ &= O((n^{r - \frac{t-\alpha}{1-\alpha} + \frac{1-t}{1-\alpha} + \epsilon})(n^{\frac{t-\alpha}{1-\alpha}})^\omega) \\ &= O(n^{r - \frac{t-\alpha}{1-\alpha} + \frac{1-t}{1-\alpha} + \frac{\omega(t-\alpha)}{1-\alpha}}) \\ &= O(n^{\frac{r(1-\alpha)+(1-t)+(\omega-1)(t-\alpha)}{1-\alpha}}) . \end{aligned}$$

□

Let $f(r,t)$ denote the right-hand side of (7.3), let $g(r, 0 \leq t \leq \alpha) = 1 + r + \epsilon$, and let

$$g(r, \alpha < t \leq 1) = \frac{r(1-\alpha) + (1-t) + (\omega - 1)(t-\alpha)}{1-\alpha} . \tag{8.3}$$

We combine these relations, and in **figure 2**, we represent the resulting exponents in this parameter range.

Part II

Applications of Algorithms for Fast Rectangular Matrix Multiplications to Accelerations of Parallel Matrix Computations and of Composition and Factorization of Polynomials over Finite Fields

9 Acceleration of Parallel Matrix Computations

In this section, we will apply the results of section 8 in order to improve the record complexity estimates of [GP89] for parallel solution of the three following problems:

1) compute the determinant and the characteristic polynomial of a given $n \times n$ rational, real, or complex matrix A;

2) solve a linear system $Ax = b$;

3) invert A.

We first repeat some basic definitions from [GP89], which are used in the main theorem and its corollary in [GP89].

Definition: $P(n)$ is the minimum number of arithmetic processors supporting $O(\log^2 n)$ parallel time bound for solving problems 1), 2) and 3) under the EREW PRAM model of parallel computing; $P(*, m, n, p)$ is the minimum number of arithmetic processors supporting $O(\log(mnp))$ parallel time bound for multiplication of $m \times n$ by $n \times p$ matrices; $P(*, n) = P(*, n, n, n)$.

The following theorem and its corollary are from [GP89]:

Theorem 9.1 *The solution to Problem 1) and 2) can be computed by using $O(\log^2 n)$ parallel steps and simultaneously*

$$P(det, n) = \max\{P(*, n^{1.25}, n, n^{1.25}), P(*, n^{0.5}, n^2, n^{0.5})\}$$

processors.

The solution to Problem 3) can be computed by using $O(\log^2 n)$ steps and

$$P(n) = \min_{v,u} \max\{P(det, n), P(*, u+1, v, n^2), P(*, n, nu, n)\}$$

processors, where the minimum is over all pairs v and u such that

$$vu \le n+1 \le (v+1)u \ .$$

Substitute the bound $P(*,n) = O(n^{2.376})$ and obtain

Corollary 9.1 *The solutions to Problems 1), 2) and 3) can be computed by using $O(\log^2 n)$ steps and $P(n) = O(n^{2.851})$ arithmetic processors.*

We will also need the following result, which extends Proposition 4.3.2 of [BP94] from the case of square to rectangular matrices:

Theorem 9.2 *The product XY of an $n^{ts} \times n^s$ matrix X by an $n^s \times n^{rs}$ matrix Y can be computed by using parallel time $O((t+r+1)s\log n)$ and $O(n^{\overline{\omega}(t,1,r)s})$ arithmetic processors, where $n > 1$, $s \to \infty$, and $\overline{\omega}(t,1,r)$ is any number exceeding the value $\omega(t,1,r)$ defined in section 2.*

Proof: With no loss of generality, we may assume (see, for instance, [BM75], section 2.5, or [Pan]) that an $n^t \times n$ by $n \times n^r$ matrix product X_0Y_0 is computed by means of a bilinear algorithm (cf. Definition 2.1).

Now we apply the tensor product construction to such a bilinear algorithm, that is, we apply this algorithm recursively in order to multiply the matrices X and Y whose entries are $n^t \times n$ and $n \times n^r$ matrices, respectively. This will give us a recursive bilinear algorithms for multiplication of $n^{ts} \times n^s$ by $n^s \times n^{rs}$ matrices, for $s = 1, 2, \cdots$, and we have

$$t_{s+1} \le t_s + (1 + \max(r,t))\log_2 n + \log_2 M + 4 \ ,$$

$$p_{s+1} \le \max\{n^{(r+t+2)(s+1)}, n^{(r+t)(s+1)}M, p_sM\} \ ,$$

where $N = n^{\max(1+r,1+t,r+t)}$, t_l and p_l denote the parallel time and the number of arithmetic processors used in the above recursive bilinear algorithm for $n^l \times n^l$ matrix multiplication.

Since $M \leq n^{\bar{\omega}(t,1,r)}$, the latter recursive relations immediately lead to Theorem 9.2. □

Next, we will apply the results of our section 8 in order to improve the bound on $P(n)$ from $O(n^{2.851})$ of [Corollary 9.1] to $O(n^{2.837})$. Due to Theorems 9.1 and 9.2, it suffices to improve the upper estimate $O(n^{2.837})$ for the sequential complexity of the four following problems of rectangular matrix multiplication

$$< n^{1.25}, n, n^{1.25} >, \quad < n^{1/3}, n^{2/3}, n^2 >, \quad < n, n^{4/3}, n >, \quad < n^{0.5}, n^2, n^{0.5} > ,$$

defined by the four following exponents:

$$\omega(1.25, 1, 1.25), \quad \omega(1/3, 2/3, 2), \quad \omega(1, 4/3, 1), \quad \omega(0.5, 2, 0.5) \ .$$

By applying the results of section 8, we obtain that

$\omega(1.25, 1, 1.25) = 1.25\omega(1, 1, 0.8) = 2.8368 \cdots < 2.837$ (by applying (8.1) for $\omega = 2.376$),

$\omega(1/3, 2/3, 2) = \frac{2}{3}\omega(0.5, 1, 3) = 2.7398 \cdots$ (by applying (8.2) for $\omega = 2.376$),

$\omega(1, 4/3, 1) = \omega(1, 1, 1.33 \cdots) = 2.6993 \cdots$ (by selecting $q = 7$, $\beta = 0.033$ in (7.1)),

$\omega(0.5, 2, 0.5) = 0.5\omega(1, 1, 4) = 2.6390 \cdots$ (by selecting $q = 14$, $\beta = 0.0026$ in (7.1)).

Combining the four latter bounds with Theorems 9.1 and 9.2, we arrive at the bound $P(n) = O(n^{2.837})$.

Remark 9.1 The bound $P(n) = O(n^{2.837})$ can be decreased if $\omega = \omega(1, 1, 1)$ is decreased below 2.376 and also if α is increased above 0.294. Namely, our argument above, together with (8.1) and (8.2) implies that

$$P(n) = O(\max\{P_1(n), P_2(n), P_3(n), P_4(n)\}) \ ,$$

where

$$P_1(n) = n^{\omega_1}, \quad \omega_1 = \omega(1.25, 1, 1.25) = 1.25\frac{0.4 + (0.8 - \alpha)\omega}{1 - \alpha} \quad \text{[cf. (8.1)]},$$

$$P_2(n) = n^{\omega_2}, \quad \omega_2 = \omega(1/3, 2/3, 2) = \frac{2}{3}\omega(0.5, 1, 3) = \left(\frac{2}{3}\right)\frac{3(1-\alpha) + 0.5 + (\omega - 1)(0.5 - \alpha)}{1 - \alpha}$$

[cf. (8.2)],

$$P_3(n) = n^{\omega_3}, \quad \omega_3 = \omega(1, 4/3, 1) < 2.7,$$

$$P_4(n) = n^{\omega_4}, \quad \omega_4 = \omega(0.5, 2, 0.5) < 2.64.$$

Clearly, ω_1 and ω_2 decrease as ω decreases and/or α increases.

10 Acceleration of Composition and Factorization of Polynomials over Finite Fields

10.1 Introduction

In this section, we will apply the results of Part I on fast rectangular matrix multiplication in order to improve the known estimates for the computational complexity of the factorization of univariate polynomials over finite fields, which is a major problem of algebraic computing. We refer the reader to [KS95] on the background of this fundamental problem and to Part I on the background of fast rectangular matrix multiplication.

10.2 Some Notations and Prerequisitic Results

In this subsection, we state some definitions and results of Part I, that we will apply in this section. Hereafter, "ops" will stand for "arithmetic operations" and "bms" will stand for bilinear multiplications. $< m, n, p >$ will denote the problem of multiplying a pair of $m \times n$ by $n \times p$ matrices. We will represent the complexity of $< n^r, n^s, n^t >$ by the number

of bilinear multiplications (bms) required, which we will denote $M(n^r, n^s, n^t)$. This is an adequate measure since

$$A(n^r, n^s, n^t) = M(n^r, n^s, n^t) \cdot n^{(r+s+t)\epsilon} \quad \text{as} \quad n \to \infty \ ,$$

provided that r, s, t are positive and $A(m, n, p)$ denotes the number of ops required for $< m, n, p >$.

Theorem 10.1 (Part I, section 5). *The problem $< n, n, n^2 >$ of Rectangular Matrix Multiplication can be solved by using $O(n^{3.333953\cdots})$ bms, that is, the exponent of the arithmetic complexity of $< n, n, n^2 >$ is $\omega(1, 1, 2) \leq 3.333953\cdots$.*

Theorem 10.2 (Part I, section 7). *The problem $< n, n, n^r >$ of Rectangular Matrix Multiplication can be solved by using $O(n^{\omega(1,1,r)})$ bms, where $r \geq 1$ is a rational number, the matrix exponent $\omega(1, 1, r)$ is bounded as follows:*

$$\omega(1, 1, r) \leq \min_{l,b} \frac{1}{(1-b)\log l} \log \left(\frac{b^b((1+r)(1-b))^{(1+r)(1-b)}(1+rb)^{(1+rb)}(l+2)^{(2+r)}}{(2+r)^{(2+r)}} \right) \ ,$$

where $l \geq 2$ is an integer and $0 \leq b \leq 1$.

Theorem 10.3 (Part I, section 8). *The problem $< n^t, n, n^r >$ of Rectangular Matrix Multiplication (where $0 \leq t \leq 0.294$, $r \geq 1$) can be solved by using $O(n^{r+1+\epsilon})$ bms.*

10.3 Complexity of Polynomial Composition over Finite Fields

Let

$$p(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n \ ,$$

$$q(x) = q_0 + q_1 x + q_2 x^2 + \cdots + q_n x^n$$

be two polynomials. Algorithm 2.1 of [BK78] for computing

$$p(q(x)) \bmod x^{n+1}$$

has complexity dominated by the complexity of the problem $< n, \sqrt{n}, \sqrt{n} >$ (cf. [BK78]), which can be reduced to $\sqrt{n}$ blocks of square matrix multiplication, so that

$$M(n, \sqrt{n}, \sqrt{n}) \leq \sqrt{n} M(\sqrt{n}, \sqrt{n}, \sqrt{n}) \ .$$

The paper [KS95] cites the record estimate for the complexity of $< n, \sqrt{n}, \sqrt{n} >$ based on this inequality and on the currently best exponent

$$\omega = \omega(1,1,1) = 2.375477 \cdots$$

for square matrix multiplication, which lead to the bound

$$M(n, \sqrt{n}, \sqrt{n}) = \sqrt{n} M(\sqrt{n}, \sqrt{n}, \sqrt{n}) = O(n^{\frac{\omega+1}{2}}) = O(n^{1.688}) \ ,$$

that is, 1.688 is the currently best exponent known for $< n, \sqrt{n}, \sqrt{n} >$. On the other hand, if the matrix exponent

$$\omega(1,1,2) < 3.333953 \cdots$$

of theorem 2.1 of section 2 for rectangular matrix multiplication problem $< n, n, n^2 >$ is employed, then

$$M(n, \sqrt{n}, \sqrt{n}) = O(n^{3.334/2}) = O(n^{1.667}) \ ,$$

which is an improvement by 0.021 over 1.688.

10.4 Complexity of Polynomial Factorization over Finite Fields

In this subsection, we will apply the results of subsection 2 for fast rectangular matrix multiplication in order to improve several complexity results of [KS95] on polynomial factorization over finite field F_q. In particular, in section 2 of [KS95], it is pointed out that

the equal-degree factorization problem in a field $\mathbf{F}_q$ can be solved on a degree n input by means of the probabilistic algorithm of von zur Gathen and Shoup (cf. [GS92]) using the expected number of

$$O(n^{(\omega+1)/2+o(1)} + n^{1+o(1)} \log q)$$

or

$$O(n^{1.688} + n^{1+o(1)} \log q)$$

operations in $\mathbf{F}_q$. Here, $n^{(\omega+1)/2+o(1)}$ is used as the estimated complexity of $< n, \sqrt{n}, \sqrt{n} >$. Due to the results of our Theorem 2.1, the bound on the complexity of the equal-degree factorization problem can be immediately improved to

$$O(n^{1.667} + n^{1+o(1)} \log q) \ .$$

Section 2 of [KS95] presents a (deterministic) algorithm (Algorithm D) for the distinct-degree factorization problem that uses

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2} + n^{1+\beta+o(1)} \log q)$$

operations in $\mathbf{F}_q$, for any β in the interval $0 \leq \beta \leq 1$ (see Theorem 2 in [KS95]).

By choosing $\omega < 2.375477$ and minimizing the exponent of n, we obtain the estimate of $O(n^{1.844} \log q)$ operations in $\mathbf{F}_q$. On the other hand, $n^{(\omega+1)/2}$ in the above estimate is a bound on the complexity of $< n, \sqrt{n}, \sqrt{n} >$. By using the results of Part I, we may replace $(\omega + 1)/2$ by $\omega(1,1,2) < 3.333953$ and minimize the exponent of n, so as to achieve the bound $O(n^{1.8335} \log q)$, which improves the known exponent by about 0.0105.

The current record complexity bound $O(n^{1.815} \log q)$ for Algorithm D is given in Theorem 3 of [KS95]. Theorem 3 relies on the fact that Algorithm D can be implemented so as to use

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2} + n^{1+\beta+o(1)} \log q)$$

operations in $\mathbf{F}_q$. If we choose $\omega < 2.375477$ and minimize the exponent of n, we will arrive at the bound of $O(n^{1.815}\log q)$ operations in $\mathbf{F}_q$. By applying the results of our theorems 10.2 and 10.3, we will improve the bound $O(n^{1.815}\log q)$ to $O(n^{1.80535}\log q)$. Doing this, we will follow the proof of Theorem 3 of [KS95]. As is pointed out in [KS95], the conclusion of Theorem 3 of [KS95] is an immediate consequence of two lemmas, that is, Lemmas 3 and 4 of [KS95]. Let us next recall and improve these lemmas.

Lemma 3 of [KS95] states: Given a polynomial $f \in \mathbf{K}[x]$ of a degree n over an arbitrary field $\mathbf{K}$ and $k+1$ polynomials $g_1,\ g_2,\cdots,\ g_k,\ h\ \in\ \mathbf{K}[x]$, all of degrees less than n, where $k = O(n^{\delta}),\ 0 \le \delta \le 1$, we can compute

$$g_1(h) \bmod f,\cdots,\ g_k(h) \bmod f\ \in\ \mathbf{K}[x]$$

by using

$$O(n^{(\omega+1)/2}k^{(\omega-1)/2})$$

operations in $\mathbf{K}$.

In the proof of Lemma 3 of [KS95], the latter complexity bound relies on the estimates for the complexity of the problem $< n, \sqrt{nk}, \sqrt{nk} >$, for which [KS95] uses the bound

$$O(n/\sqrt{nk})M(\sqrt{nk}, \sqrt{nk}, \sqrt{nk})\ .$$

We may replace this estimate by $M(n, \sqrt{nk}, \sqrt{nk})$. As is pointed out in section 8 of Part I, for most of the selections of k, our direct algorithm for rectangular matrix multiplication achieves better result than that achieved by applying the algorithm for square matrix multiplication.

Lemma 4 of [KS95] states: Let $f \in \mathbf{F}_q[x]$ be a polynomial of a degree n. Suppose that we are given $x^{q^r} \bmod f \in \mathbf{F}_q[x]$. Then we can compute

$$x^{q^r} \bmod f,\ x^{q^{2r}} \bmod f\ ,\cdots, x^{q^{Kr}} \bmod f\ \in\ \mathbf{F}_q[x]\ ,$$

for $K = O(n^{\delta})$, $0 \leq \delta \leq 1$, by using

$$O(n^{(\omega+1)/2} K^{(\omega-1)/2})$$

operations in $\mathbf{F}_q$.

For convenience, let us repeat the proof of Lemma 4 here.

Proof of Lemma 4: For $i \geq 1$, let $G_i = x^{q^{ir}} \bmod f \in \mathbf{F}_q[x]$. Assume that we have computed $G_1, \cdots, G_m$. Then we can compute $G_{m+1}, \cdots, G_{2m}$ by computing

$$G_1(G_m) \bmod f, \ \cdots \ , \ G_m(G_m) \bmod f$$

by means of the algorithm of Lemma 3. Therefore, to compute $G_1, \cdots, G_K$ given G_1, we simply repeat the above "doubling" step $O(\log K)$ times, and then achieve the stated running-time estimate.

The procedure above can be explained in the following way (so as to apply our updated version of Lemma 3):

Step 1. For a given G_1, computing $G_1(G_1) \bmod f = G_2$ is equivalent to solving the problem $< n, \sqrt{n}, \sqrt{n} >$ (i.e. $k = 1$ in Lemma 3).

Step 2. For a given G_1 and G_2 from step 1, computing

$$G_1(G_2) \bmod f = G_3 \quad \text{and} \quad G_2(G_2) \bmod f = G_4$$

is equivalent to solving $< n, \sqrt{2n}, \sqrt{2n} >$ (i.e. $k = 2$ in Lemma 3).

.....

Step $\log K - 1$. For $G_1, \cdots, G_{K/8}$ from the previous steps, computing

$$G_1(G_{K/8}) \bmod f = G_{1+K/8}, \ \cdots \ , \ G_{K/8}(G_{K/8}) \bmod f = G_{K/4}$$

is equivalent to solving $< n, \sqrt{n(K/4)}, \sqrt{n(K/4)} >$ (i.e. $k = K/4$ in Lemma 3).

Step $\log K$. For $G_1, \cdots, G_{K/4}$ from the previous steps, computing

$$G_1(G_{K/4}) \bmod f = G_{1+K/4}, \ \cdots \ , \ G_{K/4}(G_{K/4}) \bmod f = G_{K/2}$$

is equivalent to solving $< n, \sqrt{n(K/2)}, \sqrt{n(K/2)} >$ (i.e. $k = K/2$ in Lemma 3).

Since

$$M\left(n, \sqrt{n(K/2^{i+1})}, \sqrt{n(K/2^{i+1})}\right) \leq \frac{1}{2} M\left(n, \sqrt{n(K/2^i)}, \sqrt{n(K/2^i)}\right), \quad i = 1, 2, \cdots, \log K \ ,$$

by summing the complexity estimates from step 1 to step $\log K$, we arrive at the overall complexity bound of

$$\left(\frac{1}{2^{\log K}} + \cdots + \frac{1}{2}\right) M\left(n, \sqrt{nK}, \sqrt{nK}\right) < M\left(n, \sqrt{nK}, \sqrt{nK}\right) \ .$$

Therefore, we may replace $O(n^{(\omega+1)/2} K^{(\omega-1)/2})$ by $M\left(n, \sqrt{nK}, \sqrt{nK}\right)$.

According to Algorithm D, $K = n^{1-\beta}$, which leads to the result of Theorem 3 of [KS95]. Then, by replacing K by $n^{1-\beta}$ in $M(n, \sqrt{nK}, \sqrt{nK})$, we deduce the bound of

$$M(n, \sqrt{n^{2-\beta}}, \sqrt{n^{2-\beta}}) = M(n, n^{1-\beta/2}, n^{1-\beta/2}) = O(n^{\omega(1,1-\beta/2,1-\beta/2)}) \ .$$

By using the results of Lemmas 3 and 4 of [KS95], we may replace

$$n^{(\omega+1)/2+(1-\beta)(\omega-1)/2}$$

by

$$n^{\omega(1,1-\beta/2,1-\beta/2)}$$

in Theorem 3 of [KS95] and extend Theorem 3 of [KS95] so as to yield the bound of

$$O(n^{\omega(1,1-\beta/2,1-\beta/2)} + n^{1+\beta+o(1)} \log q) \ .$$

To minimize the latter exponent, we choose $\beta = 0.805347$. Furthermore, in Theorem 10.2 of subsection 2, we choose $b = 0.023$ and $l = 8$ and thus arrive at the estimate

$$\omega(1, 1-\beta/2, 1-\beta/2) \leq 1.805346859 \cdots < 1.805347 \ .$$

Since $\beta + o(1)$ is bounded from above by 0.80535, we finally arrive at the complexity bound $O(n^{1.80535} \log q)$, and this yields a new record complexity estimate for the distinct-degree factorization.

10.5 Complexity of Fast Black Box Berlekamp Algorithm

In this subsection, we will follows [KS95]. As in the preceding subsection, we will recall the results of Theorems 2 and 3 of [KS95]. By utilizing the latest results of Part I on rectangular matrix multiplication, we will improve the results of Theorems 4 and 5 of [KS95] on the complexity of the fast Black Box Berlekamp Algorithm.

First, let us recall the result of Theorem 4 of [KS95], which states that for any constant β with $0 \leq \beta \leq 1$, Algorithm B of [KS95] can be implemented so as to use an expected number of

$$O(n^{(\omega+1)/2+(3-\omega)|\beta-1/2|+o(1)} + n^{(\omega+1)/2+(1-\beta)+o(1)} + n^{1+\beta+o(1)} \log q) \qquad (10.5.1)$$

operations in $\mathbf{F}_q$. By choosing $\omega < 2.375477$ and minimizing the exponent n, we get

$$O(n^{1.880} + n^{1.808} \log q)$$

operations in $\mathbf{F}_q$, which is the bound stated in [KS95].

We will improve the latter bound to the minimum of

$$O(n^{1.860} + n^{1.808} \log q) \quad \text{and} \quad O(n^{1.8335} \log q).$$

Note first that $O(n^{1.880} + n^{1.808} \log q)$ is obtained by choosing $\beta = 0.808$; also note that $n^{(\omega+1)/2}$ comes from the complexity bound for $< n, \sqrt{n}, \sqrt{n} >$, which we will set to $O(n^{3.334/2}) = O(n^{1.667})$, due to Theorem 10.1 of subsection 2. Then, we will bound the exponents of the first and the second terms by $1.8591\cdots$ and of the third term by 1.808,

that is, we have the overall complexity bound of $O(n^{1.860} + n^{1.808} \log q)$. Next, in order to yield the estimate $O(n^{1.8335} \log q)$, we first note that it is the same as our updated results for Theorem 2 of [KS95] (see the preceding subsection) and that the second and the third terms,

$$n^{(\omega+1)/2+(1-\beta)+o(1)} + n^{1+\beta+o(1)} \log q \ ,$$

are almost same as one of Theorem 2 of [KS95]. Therefore, it suffices to prove that the exponent of the first term can also be decreased to 1.8335. For this reason, let us follow the proof of Theorem 4 of [KS95] so as to cover Step AE2 of Algorithm AE and the calculation of its complexity. The bound

$$O(n^{(\omega+1)/2+(3-\omega)|\beta-1/2|+o(1)})$$

comes from the complexity estimate for rectangular matrix multiplication problem $< m, t, n >$, where $m = n^{1-\beta}$ and $t = n^{\beta}$, or conversely, $t = n^{1-\beta}$ and $m = n^{\beta}$, that is,

$$O(n^{(\omega+1)/2+(3-\omega)|\beta-1/2|+o(1)})$$

comes from the bound

$$M(n^{1-\beta}, n^{\beta}, n) = O(n^{\omega(1-\beta,\beta,1)}) \ .$$

For $\beta = 0.8335 - o(1)$, among $1-\beta$, β, and 1, the value $1 - \beta = 0.1665 + O(1)$ is the smallest, 1 is the largest, and $(1-\beta)/\beta < .294$. By applying our Theorem 10.3, we get

$$\omega(1-\beta, \beta, 1) = 1 + \beta + o(1) = 1.8335 \ .$$

Therefore, we achieve $O(n^{1.8335} \log q)$, thus improving Theorem 4 of [KS95].

Our improvement will enable us to improve the result of Theorem 5 of [KS95].

Theorem 5 of [KS95] states: For any constant β with $0 \leq \beta \leq 1$, Algorithm B can be implemented so as to use an expected number of

$$O(n^{(\omega+1)/2+(3-\omega)|\beta-1/2|+o(1)} + n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)} + n^{1+\beta+o(1)} \log q) \qquad (10.5.2)$$

operations in $\mathbf{F}_q$.

By choosing $\omega < 2.375477$ and minimizing the exponent of n, we obtain the bound of

$$O(n^{1.852} + n^{1.763} \log q)$$

operations in $\mathbf{F}_q$. Furthermore, for $\omega = 2.375477$, by making use of the techniques for fast rectangular matrix multiplication, the estimate (10.5.2) of Theorem 5 of [KS95] can be reduced to

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)} + n^{1+\beta+o(1)} \log q) \qquad (10.5.3)$$

and, in particular, to $O(n^{1.815} \log q)$ for an appropriate choice of β.

We will respectively improve the bounds $O(n^{1.852} + n^{1.763} \log q)$ on the complexity on Algorithm B of [KS95] to $O(n^{1.8356} + n^{1.763} \log q)$ and $O(n^{1.815} \log q)$ to $O(n^{1.80535} \log q)$.

In the first case, noting that $\beta = 0.763 - o(1)$ and that the second term

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)})$$

comes from

$$M(n, n^{1-\beta/2}, n^{1-\beta/2}) = O(n^{\omega(1,1-\beta/2,1-\beta/2)})$$

(as we discussed this when we improved the result of Theorem 3 of [KS95] in the preceding section), we choose $\beta = 0.763 - o(1)$. In theorem 10.2 of subsection 2, by choosing $b = 0.023$ and $l = 8$, we arrive at

$$\omega(1, 1-\beta/2, 1-\beta/2) \leq 1.835532965 \cdots .$$

In the proof of Theorem 5 of [KS95],

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)})$$

is an upper bound on

$$M(m, n, t) = M(n^{1-\beta}, n, n^{\beta}) .$$

For $\beta = 0.763 - o(1)$, we have

$$\begin{aligned} & M(n^{1-\beta}, n, n^{\beta}) \\ &= M(n^{0.237+o(1)}, n, n^{0.763-o(1)}) \\ &= n^{0.013} M(n^{0.224+o(1)}, n, n^{0.763-o(1)}) \\ &= n^{0.013} O(n^{\omega(0.224+o(1),1,0.763-o(1))}) . \end{aligned} \tag{10.5.4}$$

On the other hand,

$$(0.224 + o(1))/(0.763 - o(1)) \leq 0.294$$

(compare the first term of (5.1) in Theorem 4 of [KS95]). Consequently, by applying Theorem 10.3 of subsection 2, we deduce that

$$\omega(0.224 + o(1), 1, 0.763 - o(1)) = 1 + 0.763 - o(1) = 1.763 .$$

Therefore,

$$n^{0.013} O(n^{\omega(0.224+o(1),1,0.763-o(1))}) = O(n^{0.013+1.763}) = O(n^{1.776})$$

bounds the first term, which is dominated by the second term. This enables us to improve the bound $O(n^{1.852} + n^{1.763} \log q)$ to $O(n^{1.8356} + n^{1.763} \log q)$.

Finally, we discuss the improvement from $O(n^{1.815} \log q)$ to $O(n^{1.80535} \log q)$. Since the second and the third terms of Theorem 5 of [KS95] [cf. (10.5.2)] are the same as in the Theorem 3 of [KS95], that is, $O(n^{1.80535} \log q)$, it remains to prove that the first term is dominated by $O(n^{1.80535})$. Noting that $\beta = 0.80535 - o(1)$ and that (as we mentioned

above) the first term comes from the bound

$$\begin{aligned} & M(m,n,t) \\ &= M(n^{1-\beta},n,n^{\beta}) \\ &= M(n^{0.19465+o(1)},n,n^{0.80535-o(1)}) \\ &= O(n^{\omega(0.19465+o(1),1,0.80535-o(1))} \\ &= O(n^{1.80535}) , \end{aligned}$$

due to the bound

$$(0.19465 + o(1))/(0.80535 - o(1)) \leq 0.294$$

and the application of our Theorem 10.3, we arrives at a new bound of $O(n^{1.80535} \log q)$, thus improving one of Theorem 5 of [KS95].

References

[BCLR] D. Bini, M. Capovani, G. Lotti and F. Romani, $O(n^{2.7799})$ complexity for matrix multiplication, *Inform. Process. Lett.*, **8**, 234-235, 1979.

[BD76] R. W. Brockett and D. Dobkin, On the Number of Multiplications Required for Matrix Multiplications, *SIAM J. on Complexity*, **5**, 4, 624-628, 1976.

[Be46] F. A. Behrend, On Sets of Integers Which Contain No Three Terms in Arithmetical Progression, *Proc. Nat. Acad. Sci. USA*, **32**, 331-332, 1946.

[BK78] R. P. Brent and H. T. Kung, Fast Algorithms for Manipulating Formal Power Series, *J. ACM*, **25**, No. 4, 581-595, October, 1978.

[BM75] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York, 1975.

[BP94] D. Bini and V. Y. Pan, *Polynomial and Matrix Computations, Vol.1: Fundamental Algorithms*, Birkhäuser Boston, 1994.

[Co82] D. Coppersmith, Rapid Multiplication of Rectangular Matrices, *SIAM J. Comput.*, **11**, No. 3, 467-471, August 1982.

[Co96] Don Coppersmith, Rectangular Matrix Multiplication Revisited, Research Report 20498, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA, 1996.

[CW81] D. Coppersmith and S. Winograd, On the Aysmptotic Complexity of Matrix Multiplication, *SIAM J. Comput.*, **11**, 472-492, 1981.

[CW90] D. Coppersmith and S. Winograd, Matrix Multiplication via Arithmetic Progressions, *J. Symb. Comp.*, **9**, 251-280, 1990.

[GP89] Z. Galil and V. Y. Pan, Parallel Evaluation of the Determinant and of the Inverse of a Matrix, *Information Proc. Letters*, **30**, 41-45, 1989.

[GS92] Joachim Von Zur Gathen and Victor Shoup, Computing Frobenius Maps and Factoring Polynomials, *Comput Complexity*, **2**, 187-224, 1992.

[KS95] Erich Kaltofen and Victor Shoup, Subquadratic-Time Factoring of Polynomials over Finite Fields, *Proc. 27th Annual ACM Symp. Theory Comp.* (New York, N.Y., 1995), ACM Press, 398-406. *Math. Comput.*, in press.

[Pan72] V. Y. Pan, On Schemes for the Computation of Products and Inverse of Matrices, *Uspekhi Mat. Nauk*, 27(5), pp.249-250, 1972. (In Russian.)

[Pan] V. Y. Pan, *How to Multiply Matrices Faster*, Lecture Notes in Computer Science, **179**, Springer, Berlin, 1984.

[Pan,a] V. Y. Pan, How Can We Speed-up Matrix Multiplication ?, *SIAM Rewiew*, **26**, 3, 393-415, 1984.

[Sc81] A. Schönhage, Partial and Total Matrix Multiplication, *SIAM J. Comput.*, **10**, 3, 434-456, 1981.

[SS42] R. Salem and D. C. Spencer, On Sets of Integers Which Contain No Three Terms in Arithmetical Progression, *Proc. Nat. Acad. Sci. USA*, **28**, 561-563, 1942.

[St69] V. Strassen, Gaussian Elimination Is Not Optimal, *Numerische Math.*, **13**, 354-356, 1969.

[St86] V. Strassen, The Asymptotic Spectrum of Tensors and the Exponent of Matrix Multiplication, *Proc. 27th Ann. IEEE Symp. on Foundations of Computer Science*, 49-54, 1986.

[St87] V. Strassen, Relative Bilinear Complexity and Matrix Multiplication, *J. reine angew. Math.*, **375/376**, 406-443, 1987.

[St88] V. Strassen, The Asymptotic Spectrum of Tensor, *J. reine angew. Math.*, **384**, 102-152, 1988.