Fast Rectangular Matrix Multiplications and Improving Parallel Matrix Computations *

Xiaohan Huang[1] and Victor Y. Pan[2]

[1] Ph.D. Program in Mathematics
Graduate School and University Center, City University of New York
33 West 42nd Street, New York, NY 10036, USA
Internet: xhuang@email.gc.cuny.edu

[2] Department of Mathematics and Computer Science
Lehman College, City University of New York
Bronx, NY 10468, USA
Internet: vpan@lcvax.lehman.cuny.edu

Abstract

Galil and Pan, 1984, reduced parallel evaluation of the inverse, the determinant and the characteristic polynomial of a matrix and solving a nonsingular linear system of equations to sequential multiplication of rectangular matrices. We asymptotically accelerate the known algorithms for the latter problem to yield an improvement of the current record asymptotic bounds on the deterministic arithmetic NC processor complexity of the four former ones, from order of $n^{2.851}$ to $O(n^{2.837})$. The improvement of rectangular matrix multiplication has also impact on the record complexity estimates for polynomial factorization in finite fields.

1 Introduction

Computing the inverse, the determinant and the characteristic polynomial of an $n \times n$ matrix and solving a nonsingular linear system of n equations are among the most fundamental problems of matrix computations. Their first NC solution was given by Csanky in his seminal paper [Cs76]. Under the EREW PRAM model of parallel computing, Csanky's algorithm can be implemented by using $O(\log^2 n)$ time and $O(n^{\omega+1})$ processors, provided that $O(n^{\omega})$ arithmetic operations suffice in order to multiply a pair of $n \times n$ matrices (current record bounds are $2 \leq \omega < 2.37547\cdots$ [CW90]). The processor bound was later improved first to $O(n^{\omega+0.5})$ [PS78] and then to $O(n^{\omega+0.5-\delta(\omega)})$ for a positive $\delta(\omega)$ [GP89].

*Supported by NSF Grant CCR 9625344 and PSC CUNY Award 667340.

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. PASCO'97, Wailea, Maui, Hawaii; ©1997 ACM 0-89791-951-3/97/0007...US$3.50

Our present work was motivated by the paper [GP89], which reduced the parallel solution of the cited fundamental problems of matrix computations to sequential rectangular matrix multiplication. Based on the known results for the latter problem, Galil and Pan reached the processor bound $O(n^{2.851})$ for the former problems. In our present paper, we focused on the improvement of the known solution algorithms for sequential rectangular matrix multiplication by means of extending the effective techniques of [CW90] developed for square matrix multiplication.

This enabled to yield the desired improvements of both sequential rectangular matrix multiplication and, therefore, the parallel complexity bounds for matrix computations, for which we support the time bound $O(\log^2 n)$ by using $O(n^{2.837})$ arithmetic processors. Our improvement of rectangular matrix multiplication has further theoretical applications, in particular, to improving the known estimates for the complexity of the factorization of a univariate polynomial over finite fields, which we will report in a separate paper. Like [CW90], we deal with algorithms that only give asymptotic improvement of the known complexity bounds for very large n, so the algorithms are not assumed to be practically used. Our complexity bounds are deterministic; they can be improved substantially, to the level of n^{ω} for processors, by using randomization (see [P87], [KP91], [KP92], [KP94], [BP94], and [P96]).

2 Definitions and Some Background

In this section, we will first recall the definitions and 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.

Definition 2.1: $P(n)$ 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 2.1 *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 2.1 *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.*

Now, we are motivated to seek some improvements of the known algorithms for rectangular matrix multiplication. We will first recall some definitions and background material.

Given a pair of $m \times n$ and $n \times p$ matrices $X = [x_{i,j}]$ and $Y = [y_{j,k}]$, the problem of computing the $m \times p$ matrix XY is called $m \times n$ by $n \times p$ matrix multiplication and is denoted $< m,n,p >$. Here and hereafter, we will assume that the subscripts i, j, and k range from 0 to $m-1$, $n-1$, and $p-1$, respectively. Hereafter, $A(m,n,p)$ denote the minimum number of arithmetic operations required for $< m,n,p >$.

Definition 2.2: *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.*

Let $M(m,n,p)$ denote the minimum M in all bilinear algorithms for $< m,n,p >$. Hereafter, we will focus on estimating $M(m,n,p)$ from above, motivated by the following known bound (cf. e.g. [Pan]):

$$A(m^h,n^h,p^h) = O((M(m,n,p))^h) \ \text{ as } \ h \to \infty\ . \tag{2.3}$$

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

$$M(m,n,1) = mn\ , \tag{2.4}$$

$$M(m,n,p) \le M(m/q,n/q,p/q)M(q,q,q) \tag{2.5}$$

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

$$\begin{array}{rcccccc} M(m,n,p) & = & M(m,p,n) & = & M(n,p,m) \\ = M(n,m,p) & = & M(p,n,m) & = & M(p,m,n) \end{array} \tag{2.6}$$

(cf. [P72] or [CW81]);

$$M(n,n,r(n)) = n^2 + o(n)$$

if $r(n) = o(\log n)$, $n \to \infty$ (cf. [BD76]);

$$A(n,n,n^r) = O(n^{2+\epsilon})$$

for any $\epsilon > 0$ if $r \le 0.197$, $n \to \infty$ (cf. [Co82]);

$$A(n,n,n^r) = O(n^{2+\epsilon})$$

for any $\epsilon > 0$ if $r \le 0.294$, $n \to \infty$ (cf. [Co]).

By extending (2.5), we obtain that

$$\begin{array}{rcl} M(m,n,p) & = & O(q^\omega)\max(mn,np,pm)/q^2, \\ q & = & \min(m,n,p) \to \infty\ , \end{array}$$

provided that $M(q,q,q) = O(q^\omega)$.

Let us next recall or introduce some basic concepts and definitions concerning matrix multiplication and recall some basic results.

The notation $L \to < m,n,p >$ 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 $L \overset{\lambda}{\to} < m,n,p >$. 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 paper, 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 (2.6), we have

$$\begin{array}{rcccccc} \omega(r,s,t) & = & \omega(t,r,s) & = & \omega(s,t,r) \\ = \omega(r,t,s) & = & \omega(s,r,t) & = & \omega(t,s,r). \end{array} \tag{2.7}$$

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:

$$\omega(r,s,t) \geq \max\{r+s, s+t, t+r\} . \qquad (2.8)$$

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

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

which matches the lower bound $\omega(1,1,t/r) \geq 2$ of (2.8), 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 2.2 (Schönhage [Sc81]) *Assume given a field F, coefficients $\alpha_{i,j,h,l}$, $\beta_{j,k,h,l}$, $\gamma_{k,i,h,l}$ in $F(\lambda)$ (the field of rational functions in a single indeterminate λ), and polynomials f_g over 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 2.2 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 2.3 (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 2.3 in order to transform tensor product construction into the form $k \cdot < m,n,p >$ for sufficiently large k, m, n and p.

Due to the application to parallel computing, we will also need the following result, which extends Proposition 4.3.2 of [BP94] ¿from the case of square to rectangular matrices:

Theorem 2.4 *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 above.*

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 $X_0 Y_0$ is computed by means of a bilinear algorithm (cf. our Definition 2.2).

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

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

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

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 $M \leq n^{\overline{\omega}(t,1,r)}$, the latter recursive relations immediately lead to Theorem 2.4. □

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 [P72] (cf. also [Pan] and [Pan,a]). For a given value of the integer q, we will call this construction D_q.

$$\sum_{i=1}^{q} \lambda^{-2} (x_0^{[0]} + \lambda x_i^{[1]})(y_0^{[0]} + \lambda y_i^{[1]})(z_0^{[0]} + \lambda z_i^{[1]})$$

$$- \lambda^{-3} (x_0^{[0]} + \lambda^2 \sum_{i=1}^{q} x_i^{[1]})(y_0^{[0]} + \lambda^2 \sum_{i=1}^{q} y_i^{[1]}) \times$$

$$(z_0^{[0]} + \lambda^2 \sum_{i=1}^{q} z_i^{[1]}) + [\lambda^{-3} - q\lambda^{-2}](x_0^{[0]})(y_0^{[0]})(z_0^{[0]})$$

$$= \sum_{i=1}^{q} (x_0^{[0]} y_i^{[1]} z_i^{[1]} + x_i^{[1]} y_0^{[0]} z_i^{[1]} + x_i^{[1]} y_i^{[1]} z_0^{[0]}) + O(\lambda). \qquad (3.1)$$

The x-variables in (3.1) consist of two blocks:

$$X^{[0]} = \{x_0^{[0]}\} \quad \text{and} \quad X^{[1]} = \{x_1^{[1]}, \cdots, x_q^{[1]}\}.$$

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

$$X_{q\times q}Y_{q\times q^2}Z_{q^2\times q}$$

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 Estimating 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

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

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 3.323953 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):

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

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

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

non-zero triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$, which share no variables with each other, where $M' \ge M^{1-\epsilon}$, 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

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

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

$$(q+2)^{4N} \ge \left[\frac{256}{\beta^\beta(3-3\beta)^{(3-3\beta)}(1+2\beta)^{(1+2\beta)}}\right]^N \times$$

$$(c')^{N\epsilon} c N^{-1+\frac{3}{2}\epsilon} q^{N(1-\beta)\omega(1,1,2)} .$$

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

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

$$\omega(1,1,2) \le \frac{1}{(1-\beta)\log q} \times$$

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

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

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

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 multiplications 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 $0 \le t \le 1$;
(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 $r = s \ge t$. Then, by homogeneity of the exponent,

$$\omega(r,s,t) = r\omega(1,1,t/r), \quad 0 \le t/r \le 1.$$

(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,

$$\omega(r,s,t) = s\omega(r/s, 1, t/s), \quad r/s > 1 > t/s > 0.$$

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

Due to (2.6), 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

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

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

$$M = 2\binom{2N}{N, N} + 1,$$

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

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

for $0 \leq r \leq 1$.

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

Due to (2.6), 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 $\binom{(1+r)N}{1N, rN}$ 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

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

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

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

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

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

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

We select the larger former bound and set

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

Construct a Salem-Spencer set B. Select random integers $0 \le w_j < M$, $j = 0, 1, 2, \cdots, (2+r)N$. As before, we obtain at least

$$\widehat{H} = \frac{1}{4} \frac{M'}{M^2} \binom{(2+r)N}{L, L, rL, r(N-L), (N-L), (N-L)}$$

non-zero triples $(X^{[I]}, Y^{[J]}, Z^{[K]})$, which share no variables with each other, where M' is the cardinality of B and $M' \ge M^{1-\epsilon}$. 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

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

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

$$(q+2)^{(2+r)N} \ge$$

$$\left[\frac{(2+r)^{(2+r)}}{\beta^\beta ((1+r)(1-\beta))^{(1+r)(1-\beta)} (1+r\beta)^{(1+r\beta)}} \right]^N \times$$

$$(c')^{\epsilon N} c N^{-1+\frac{3}{2}\epsilon} q^{N(1-\beta)\omega(1,1,r)} .$$

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

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

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

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

$$\left. \times \beta^\beta (1+r\beta)^{(1+r\beta)} (q+2)^{(2+r)} \right) . \qquad (7.1)$$

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

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

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

$$\left. (r(1-\beta)+2\beta)^{(r(1-\beta)+2\beta)} (q+2)^{(2+r)} \right) . \qquad (7.2)$$

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

Due to (2.6), 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

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

8 Discussion on Optimization

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 $r \ge 1.171 - \epsilon$ for an appropriate small positive ϵ.

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

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 [Co], 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 (2.8).]

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

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

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

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

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

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

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

□

Let $f(r,t)$ denote the right-hand side of (7.3), let

$$g(r, 0 \le t \le \alpha) = 1 + r + \epsilon,$$

and let

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

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

9 Improved Complexity of Parallel Evaluation of the Determinant and of the Inverse of a Matrix

In this section, 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 2.1] to $O(n^{2.837})$. Due to Theorems 2.1 and 2.4, it suffices to improve the upper estimate $O(n^{2.837})$ for the sequential complexity of the four following problems of rectangular matrix multiplication

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

defined by the four following exponents:

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

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 $\omega = 2.376$);

$$\omega(1/3,2/3,2) = \frac{2}{3}\omega(0.5,1,3) = 2.7398073\cdots$$

(by applying (8.2) for $\omega = 2.376$);

$$\omega(1,4/3,1) = \omega(1,1,1.33\cdots) = 2.699318\cdots$$

(by selecting $q = 7$, $\beta = 0.033$ in (7.1));

$$\omega(0.5,2,0.5) = 0.5\omega(1,1,4) = 2.6390965\cdots$$

(by selecting $q = 14$, $\beta = 0.0026$ in (7.1)).

Combining the four latter bounds with Theorems 2.1 and 2.4, we arrive at the bound $P(n) = O(n^{2.837})$.

10 Discussion

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

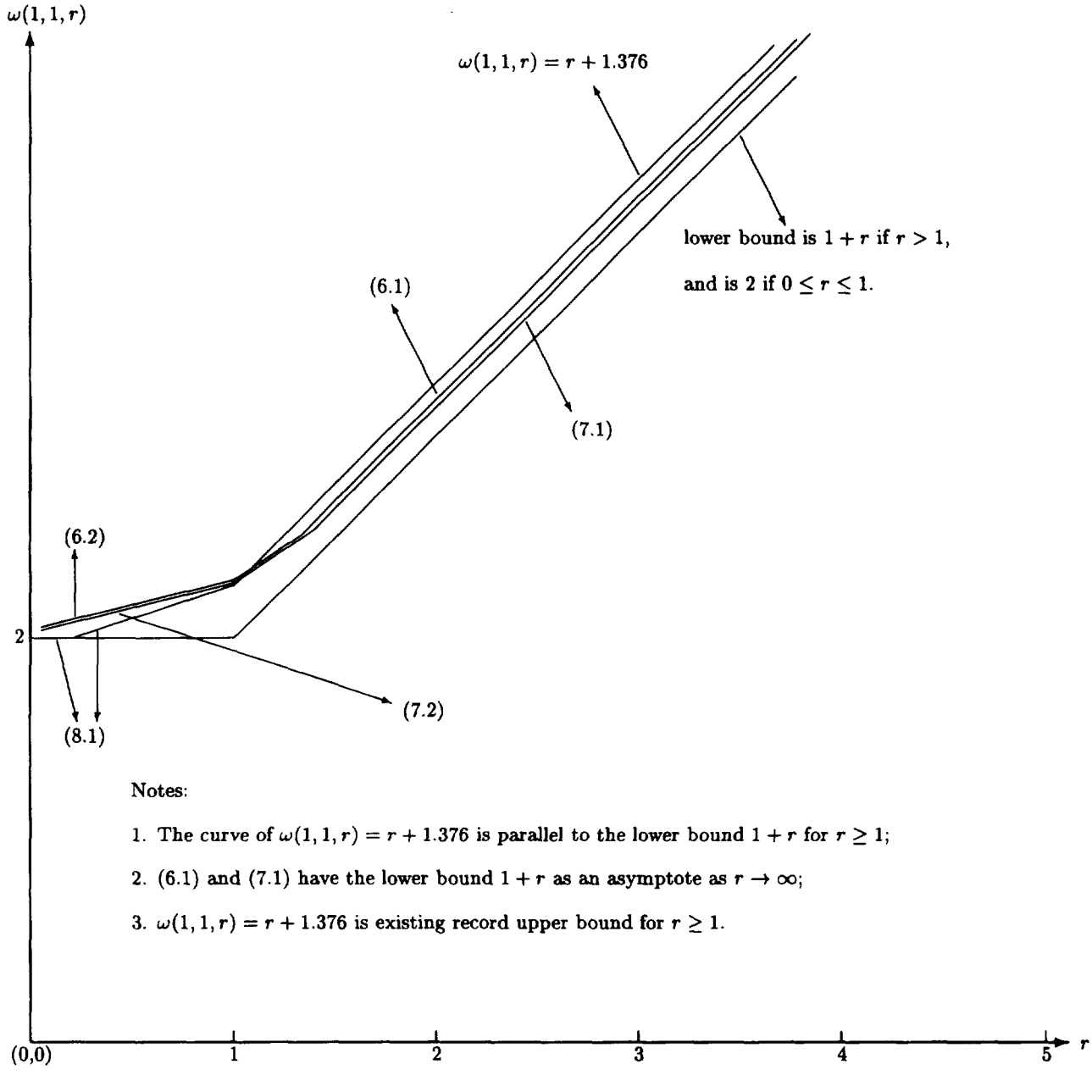
$$\begin{aligned} P_1(n) &= n^{\omega_1}, \\ \omega_1 &= \omega(1.25,1,1.25) \\ &= 1.25^{\frac{0.4+(0.8-\alpha)\omega}{1-\alpha}} \quad \text{[cf. (8.1)]}; \end{aligned}$$

$$\begin{aligned} P_2(n) &= n^{\omega_2}, \\ \omega_2 &= \omega(1/3,2/3,2) = \tfrac{2}{3}\omega(0.5,1,3) \\ &= \left(\tfrac{2}{3}\right)^{\frac{3(1-\alpha)+0.5+(\omega-1)(0.5-\alpha)}{1-\alpha}} \quad \text{[cf. (8.2)]}; \end{aligned}$$

$$P_3(n) = n^{\omega_3}, \quad \omega_3 = \omega(1,4/3,1) < 2.7;$$
$$P_4(n) = n^{\omega_4}, \quad \omega_4 = \omega(0.5,2,0.5) < 2.64.$$

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

Smaller processor bounds, at the optimum level of n^ω (still supporting polylogarithmic time bound), have been obtained for the cited fundamental matrix computations by using randomization [P87], [KP91], [KP92], [KP94], [BP94], [P96]. Is it possible to improve our NC deterministic processor bounds to the optimal level without using randomization?



Notes:

1. The curve of $\omega(1,1,r) = r + 1.376$ is parallel to the lower bound $1 + r$ for $r \geq 1$;
2. (6.1) and (7.1) have the lower bound $1 + r$ as an asymptote as $r \to \infty$;
3. $\omega(1,1,r) = r + 1.376$ is existing record upper bound for $r \geq 1$.

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

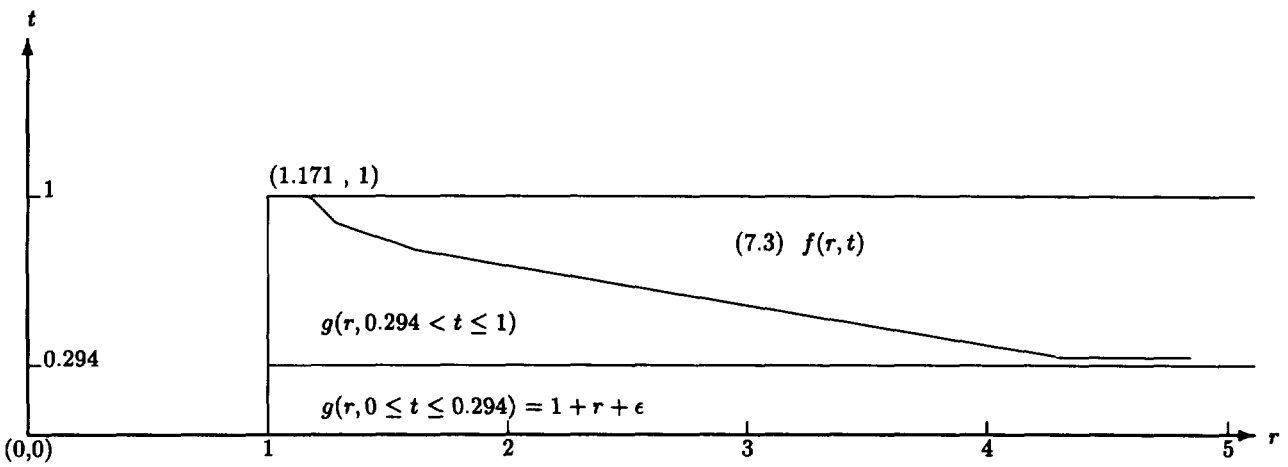


Figure 2: the three areas are the optimal region of the three exponent functions for $< n^t, n, n^r >$, $0 \leq t \leq 1 \leq r$, respectively.

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.

[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**, 3, 467-471, 1982.

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

[Cs76] L. Csanky, Fast Parallel Matrix Inversion Algorithm, *SIAM J. Computing*, **5**, 4, 618-623, 1976.

[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.

[KP91] E. Kaltofen and V. Y. Pan, Processor Efficient Parallel Solution of Linear Systems over an Abstract Field, *Proc. of 3rd Ann. ACM Symp. on Parallel Algorithms and Architectures*, 180-191, ACM Press, New York, 1991.

[KP92] E. Kaltofen and V. Y. Pan, Processor Efficient Parallel Solution of Linear Systems II. The Positive Characteristic and Singular Cases, *Proc. of 33rd Ann. IEEE Symp. on Foundations of Computer Science*, 714-723, IEEE Computer Society Press, 1992.

[KP94] E. Kaltofen and V. Y. Pan, Parallel Solution of Toeplitz and Toeplitz-like Linear Systems over Fields of Small Positive Characteristic, *Proc. of 1st Intern. Symp. on Parallel Symbolic Computation (PASCO'94)*, Linz, Austria (Sept. 1994), Lecture Notes Series in Computing, **5**, 225-233, World Scientific Publishing Company, Singapore, 1994.

[P72] V. Y. Pan, On Schemes for the Computation of Products and Inverse of Matrices, *Uspekhi Mat. Nauk*, **27**, 5, 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.

[P87] V. Y. Pan, Complexity of Parallel Matrix Computations, *Theoretical Computer Science*, **54**, 65-85, 1987.

[P96] V. Y. Pan, Parallel Computation of Polynomial GCD and Some Related Parallel Computations over Abstract Fields, *Theoretical Computer Science*, **162**, 2, 173-223, 1996.

[PS78] F. P. Preparata and D. V. Sarwate, An Improved Parallel Processor Bound in Fast Matrix Inversion, *Inform. Proc. Letters*, **7**, 3, 148-149, 1978.

[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 Progresion. *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.