JOURNAL OF COMPLEXITY **14**, 257–299 (1998)
ARTICLE NO. CM980476

Fast Rectangular Matrix Multiplication and Applications*

Xiaohan Huang

Ph.D. Program in Mathematics, Graduate School and University Center, City University of New York, 33 West 42nd Street, New York, New York 10036
E-mail: xhuang@email.gc.cuny.edu

and

Victor Y. Pan

Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, New York 10468
E-mail: vpan@lcvax.lehman.cuny.edu

Received January 21, 1997

First we study asymptotically fast algorithms for rectangular matrix multiplication. We begin with new algorithms for multiplication of an $n \times n$ matrix by an $n \times n^2$ matrix in arithmetic time $O(n^\omega)$, $\omega = 3.333953...$, which is less by 0.041 than the previous record 3.375477.... Then we present fast multiplication algorithms for matrix pairs of arbitrary dimensions, estimate the asymptotic running time as a function of the dimensions, and optimize the exponents of the complexity estimates. For a large class of input matrix pairs, we improve the known exponents. Finally we show three applications of our results:

(a) we decrease from 2.851 to 2.837 the known exponent of the work bounds for fast deterministic (NC) parallel evaluation of the determinant, the characteristic polynomial, and the inverse of an $n \times n$ matrix, as well as for the solution to a non-singular linear system of n equations,

(b) we asymptotically accelerate the known sequential algorithms for the univariate polynomial composition mod x^n, yielding the complexity bound $O(n^{1.667})$ versus the old record of $O(n^{1.688})$, and for the univariate polynomial factorization over a finite field, and

* This work was supported by NSF Grant 9625344 and PSC CUNY Awards 667340 and 668365. Some results of this paper have been presented at the Second ACM International Symposium on Parallel Algebraic and Symbolic Computations (PASCO'97), Maui, Hawaii, July 1997.

0885-064X/98 $25.00

(c) we improve slightly the known complexity estimates for computing basic solutions to the linear programming problem with m constraints and n variables.

Key Words: rectangular matrix multiplication; asymptotic arithmetic complexity; bilinear algorithms; parallel complexity; polynomial composition; polynomial factorization over finite fields; linear programming.

1. INTRODUCTION

1.1. *Our Subject and Results*

Acceleration of matrix multiplication is a major subject of theory and practice of computing (see [Pan], [Pan,a], [CW90], [GL96]). In some respects this is a basic problem in the study of computational complexity, because a very large class of computations with matrices, graphs, and regular and Boolean expressions can be reduced to matrix multiplication, so that the estimates for the asymptotic complexity of all these computations are represented by the same exponent as matrix multiplication [Pan], [BP94], [BCS97]. In this large class there is a subclass of important computational problems whose record asymptotic complexity is reduced to that of rectangular matrix multiplication. This motivates our study in the present paper, in which we improve the known upper estimates for the asymptotic complexity of multiplying rectangular matrices of large sizes and demonstrate further impact of our results. In particular, this impact includes the improvement of the known deterministic asymptotic upper bounds on the work-complexity of some of the most fundamental parallel (NC) matrix computations, such as the evaluation of the determinant, the inverse, and the characteristic polynomial of an $n \times n$ matrix as well as for the solution of a nonsingular system of n linear equations. Furthermore, we decrease the known asymptotic complexity estimates for polynomial composition, factorization of univariate polynomials over finite fields, and computation of a basis solution to a linear programming problem.

Our progress relies on extending the powerful techniques of [CW90] for fast multiplication of square matrices to rectangular matrix multiplication and on the reduction of other listed computational problems (of parallel matrix computation, polynomial composition and factorization, and linear programming) to rectangular matrix multiplication. Our techniques of the reduction of polynomial computations to rectangular matrix multiplication may be also of some independent interest because matrix computations on present day computers are known to be highly efficient [GL96], and the reduction to them is a practical means of improving the known solution of other computational problems.

As in [CW90], as well as in [Sc81], [CW82], [Co82], [Pan], [Pan,a], [St86], [St87], [St88], [GP89], [BCS97], [Co97], [BM98], we study the improvement of the known arithmetic complexity estimates for the operations with matrices of very large sizes, which are far beyond the sizes encountered in practice, and our improvement is expressed in terms of decreasing the exponents β of the complexity bounds of the form $O(N^\beta)$, $N \to \infty$, N representing the size of the input. In particular the known complexity estimate for multiplication of an $n \times n$ matrix by an $n \times n^2$ matrix was $O(n^{3.375477\ldots})$ (based on the straightforward application of [CW90]), and we decreased this exponent by roughly 0.04, to yield $O(n^{3.333953\ldots})$. For the cited problems of parallel NC computations for $n \times n$ input matrices, the known estimate for their work-complexity was $O(n^{2.851})$ [GP89], and we yielded $O(n^{2.837})$. For polynomial composition modulo x^n, we decreased the known sequential complexity exponent from $O(n^{1.688})$ (obtained by combining [BK78] and [CW90]) to $O(n^{1.667})$ with the respective decrease of the asymptotic complexity of the known fast algorithms [GS92], [KS95] for factorization of univariate polynomials in finite fields; furthermore, we showed some additional ways to improve the factorization by its more effective reduction to rectangular matrix multiplication (see the details in Section 10). Finally, application of our fast algorithms for rectangular matrix multiplication immediately enabled us to improve the estimate $O(m^{1.594}n)$ of [BM98], to yield $O(m^{1.575}n)$, for computing basic solutions to the linear programming problem with m constraints and n variables.

1.2. *Some Related Work*

Asymptotic arithmetic complexity of square $n \times n$ matrix multiplication has been studied very extensively and intensively for many years (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 dramatic improvement over the classical $\omega = 3$ (before 1969), but still falls short of the best lower bound 2.

Less attention has been paid so far to the complexity of rectangular matrix multiplication, where the most important works are [BD76], [Co82] and [Co97]. The papers [BK78], [GP89], [GS92], [KS95], [BM98] also studied the applications of rectangular matrix multiplication to the computational problems that we consider in our present paper.

1.3. *Organization of Our Paper*

We organize our presentation as follows. In Section 2, we recall some basic concepts, definitions and results on matrix multiplications. In particular, we introduce the notation $\langle m, n, p \rangle$ for the problem of $m \times n$ by

$n \times p$ matrix multiplication. In Sections 3 and 4, we modify slightly the technique of Section 6 of [CW90], which gives us an algorithm for $\langle n, n, n^2 \rangle$ 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], to improve our algorithm for $\langle n, n, n^2 \rangle$ and to yield the bound $O(n^{3.33396})$. In Section 6, we show a basic fast algorithm for $\langle n^t, n, n^r \rangle$ for an arbitrary pair of non-negative rational numbers t and r, which we improve further in Section 7. 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. We extend our improvement of rectangular matrix multiplication to the improvement of the known upper estimates for the work-complexity of deterministic parallel matrix computations in Section 9, for polynomial composition and univariate polynomial factorization over finite fields in Section 10, and for finding basic solutions to the linear programming problem in Section 11.

PART I. ACCELERATION OF RECTANGULAR MATRIX MULTIPLICATION

2. DEFINITIONS AND SOME BACKGROUND

In this section, we introduce some basic concepts and definitions concerning matrix multiplication, define some new concepts, and recall some basic results.

The problem of multiplying an $m \times n$ matrix by an $n \times p$ matrix is denoted $\langle m, n, p \rangle$. Indices i, j, k range from 0 to $m-1$, $n-1$, $p-1$, respectively.

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.

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: Evaluate first 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 $q = 0, 1, ..., M-1$, 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 $i = 0, 1, ..., m-1$; $j = 0, 1, ..., n-1$; $k = 0, 1, ..., p-1$. 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 minimum number $M(m, n, p)$ of bilinear multiplications used in all bilinear algorithms for $m \times n$ by $n \times p$ matrix multiplication, $\langle m, n, p \rangle$, is an appropriate measure for the asymptotic complexity of $\langle m, n, p \rangle$ due to the known bound (cf. e.g., [Pan])

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

In addition, 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 simple known estimates (cf. e.g., [Pan])

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

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

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

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

of [Pan72],

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

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

$$A(n, n, n^r) = O(n^{2+\varepsilon}) \qquad \text{for any } \varepsilon > 0 \text{ if } r \leqslant 0.294, \quad n \to \infty \quad [\text{Co97}].$$

By extending (2.5), we obtain that

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

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

Hereafter, the notation $L \to \langle m, n, p \rangle$ 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 $L \xrightarrow{\lambda} \langle m, n, p \rangle$. If k disjoint matrix products of the size $\langle m, n, p \rangle$ are computed (sharing no variables), we write $L \to k\langle m, n, p \rangle$.

In this paper, we study the problems of matrix multiplication of the form $\langle n^r, n^s, n^t \rangle$ 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 $\langle n^r, n^s, n^t \rangle$, that is, $O(n^{\omega(r, s, t)})$ bilinear multiplications suffice for solving the problem $\langle n^r, n^s, n^t \rangle$. Then $\omega(r, s, t)$ will be called the exponent for the problem $\langle n^r, n^s, n^t \rangle$. Due to (2.6), we have

$$\omega(r, s, t) = \omega(t, r, s) = \omega(s, t, r) = \omega(r, t, s) = \omega(s, r, t) = \omega(t, s, r). \tag{2.7}$$

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

Since

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

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

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

There is the straightforward (information) lower bound:

$$\omega(r, s, t) \geqslant \max\{r + s, s + t, t + r\}. \tag{2.8}$$

If $r = s = t$, then $\langle n^r, n^s, n^t \rangle = \langle n^r, n^r, n^r \rangle$ represents a square matrix multiplication. 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 $r = s \neq t$, then $\langle n^r, n^s, n^t \rangle$ represents multiplication of a square matrix by a rectangular matrix. Computing its bilinear complexity is reduced to

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

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

matches the lower bound $\omega(1, 1, t/r) \geqslant 2$ of (2.8), up to the term $o(1)$.

In this paper, we study the problem $\langle n^r, n^s, n^r \rangle$ of multiplication of a rectangular matrix by a rectangular matrix, where r, s and t are distinct from each other or at least $s \neq r$.

We will use 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* $\mathbf{F}^3$, *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* $\varepsilon > 0$, *one can construct an algorithm to multiply* $N \times N$ *square matrices in* $O(N^{3\tau+\varepsilon})$ *operations, where* τ *satisfies*

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

A simple extension of Theorem 2.1 enables us to estimate $\omega(r, s, t)$ from above as soon as we obtain a bilinear algorithm for k disjoint problems $\langle n^r, n^s, n^t \rangle$.

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

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

In part of our presentation, we will 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 $k\langle m, n, p\rangle$ for sufficiently large k, m, n and p.

Remark 2.1. Our study of matrix multiplication applies to the computation over arbitrary field of constants.

3. BASIC ALGORITHM FOR $\langle n, n, n^2\rangle$

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], Eq. (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:

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

The x-variables in (3.1) consist of two blocks: $X^{[0]}=\{x_0^{[0]}\}$ and $X^{[1]}=\{x_1^{[1]}, ..., x_q^{[1]}\}$. 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 $\{0, 1, 2, ..., q\}$. 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. Similar comments apply to the y- and z-variables.

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 $\langle m, n, p\rangle$ 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 $\langle q^N, q^N, q^{2N}\rangle$ 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, $\binom{Q}{Q_1, Q_2, ..., Q_s}$, for positive integers $Q, Q_1, Q_2, ..., Q_s$ satisfying

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

denotes 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 $X^{[10110100]}Y^{[11011011]} \times Z^{[01101111]}$ 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 $M' \geqslant M^{1-\varepsilon}$. 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

$$\langle q^N, q^N, (q^N)^2 \rangle.$$

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

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

which represents the matrix product

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

We deduce from the above algorithm and from Theorem 2.2 and extended Theorem 2.1 that

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

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

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

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

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

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

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

$$\omega(1,1,2) \leqslant 3.339848783... < 3.3399. \tag{3.6}$$

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 $j=0, 1, 2, ..., 4N$, and compute the integers

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

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

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

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

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

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

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

and therefore,

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

by the virtue of the Salem-Spencer 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 pattern $\langle n^r, n^s, n^t \rangle$ 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, $\langle n^r, n^s, n^t \rangle = \langle n, n, n^2 \rangle$, 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

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

respectively). Determine the largest of them (above, 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 $\langle n, n, n^2 \rangle$

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 trilinear aggregating algorithm from [CW90], Eq. (10):

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

The subscripts now form three classes: $\{0\}$, $\{q+1\}$ and $\{1, 2, ..., q\}$, 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 $4N$ $\{0, 1, 2, ..., q, q+1\}$, formed by the $4N$ subscripts j. Its superscripts $[I]$ 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

$$\begin{aligned}
&x_0^{[0]} y_i^{[1]} z_i^{[1]} && \text{from } 2N-2L \text{ copies,}\\
&x_i^{[1]} y_0^{[0]} z_i^{[1]} && \text{from } N-L \text{ copies,}\\
&x_i^{[1]} y_i^{[1]} z_0^{[0]} && \text{from } N-L \text{ copies,}\\
&x_0^{[0]} y_0^{[0]} z_{q+1}^{[2]} && \text{from } L \text{ copies,}\\
&x_0^{[0]} y_{q+1}^{[2]} z_0^{[0]} && \text{from } L \text{ copies and}\\
&x_{q+1}^{[2]} y_0^{[0]} z_0^{[0]} && \text{from } 2L \text{ copies.}
\end{aligned}$$

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 $0 \leqslant w_j < M$, $j = 0, 1, 2, ..., 4N$. 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 $M' \geqslant M^{1-\varepsilon}$, for a fixed positive ε, is the cardinality of B. Each of these triples corresponds to a matrix product of size

$$\langle q^{N-L}, q^{N-L}, (q^{N-L})^2\rangle,$$

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

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

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

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

Let $\varepsilon \to 0$, $N \to \infty$, take Nth roots and then logarithms on both sides and deduce that

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

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

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

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

6. BASIC ALGORITHM FOR $\langle n^r, n^s, n^t \rangle$

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 $\langle n^r, n^s, n^t \rangle$. For convenience, we first classify the triples $\langle n^r, n^s, n^t \rangle$, for all rational r, s, t as follows:

(1) $\langle n^r, n, n \rangle$ with $r > 1$;

(2) $\langle n, n, n^t \rangle$ with $0 \leqslant t \leqslant 1$;

(3) $\langle n^r, n, n^t \rangle$ 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. (2.7)]. 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 $r = s \geqslant t$. Then, by homogeneity of the exponent, $\omega(r, s, t) = r\omega(1, 1, t/r)$, $0 \leqslant t/r \leqslant 1$.

(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 of* $\langle n^r, n, n \rangle$ *with* $r > 1$

We begin with the construction (3.1) again. Take the $(2+r)N$th tensor power of (3.1), where N is sufficiently large and $(2+r)N$ 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 $\{0, 1, 2, ..., q\}$, 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 $\langle q^N, q^N, (q^N)^r \rangle$, 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 $M' \geqslant N^{1-\varepsilon}$. 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

$$\langle q^N, q^N, (q^N)^r \rangle,$$

which is $\langle n, n, n^r \rangle$ for $q^N = n$. It follows that

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

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

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

where c and c' are constants. Let $\varepsilon \to 0$, $N \to \infty$, take Nth roots, and obtain

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

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

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

6.2. *The Case of* $\langle n, n, n^t \rangle$ *with* $0 \leqslant t \leqslant 1$

We replace t by r, for convenience. In this case the algorithm is almost completely the same as in the case of $\langle n^r, n, n \rangle$ with $r > 1$. 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

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

for $0 \leqslant r \leqslant 1$.

6.3. *The Case of* $\langle n^r, n, n^t \rangle$ *with* $r > 1 > t > 0$

Due to (2.6), we may assume $\langle n^t, n, n^r \rangle$ with $r > 1 > t > 0$, instead of $\langle n^r, n, n^t \rangle$ 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 and $(t+1+r)N$ 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 $\langle (q^N)^t, q^N, (q^N)^t \rangle$ 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 $s_k^{[K]} = 0$ unless K has exactly N indices of 0 and exactly $(t+1)N$ 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

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

7. IMPROVED ALGORITHM FOR $\langle n^r, n^s, n^t \rangle$

In this section, we will improve our algorithm of Section 6 for the problem $\langle n^r, n^s, n^t \rangle$ 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 of* $\langle n, n, n^r \rangle$ *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 and $(2+r)N$ 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 $\{0, 1, 2, ..., q, q+1\}$, 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

$$\begin{aligned}
&x_0^{[0]} y_i^{[1]} z^{[1]} && \text{from } r(N-L) \text{ copies,}\\
&x_i^{[1]} y_0^{[0]} z_i^{[1]} && \text{from } (N-L) \text{ copies,}\\
&x_i^{[1]} y_i^{[1]} z_0^{[0]} && \text{from } (N-L) \text{ copies,}
\end{aligned}$$

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

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

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

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

$$0 \leqslant w_j < M, \qquad j = 0, 1, 2, \ldots, (2+r)\, N.$$

As before, we obtain at least

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

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

$$\langle q^{(N-L)}, q^{(N-L)}, q^{r(N-L)} \rangle.$$

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

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

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

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

Letting $\varepsilon \to 0$, $N \to \infty$, and taking Nth roots, we obtain

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

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

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

7.2. *The Case of* $\langle n, n, n^r \rangle$ *with* $0 \leqslant r \leqslant 1$

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 preceding subsection, we obtain the exponent bound

$$\omega(1,1,r) \leqslant \frac{1}{(1-\beta)\log q} \times \log\left(\frac{\left(\begin{array}{c}(r\beta)^{(r\beta)}\,(2(1-\beta))^{2(1-\beta)} \\ \times (r(1-\beta)+2\beta)^{(r(1-\beta)+2\beta)}\,(q+2)^{(2+r)}\end{array}\right)}{(2+r)^{(2+r)}}\right). \tag{7.2}$$

7.3. *The Case of* $\langle n^r, n, n^t\rangle$ *with* $r>1>t>0$

Due to (2.6), we will discuss the problem $\langle n^t, n, n^r\rangle$ with $r>1>t>0$, instead of $\langle n^r, n, n^t\rangle$ 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, and $(t+1+r)N$ 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 $\{0, 1, 2, ..., q, q+1\}$, 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 to 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

$$
\begin{aligned}
&x_0^{[0]} y_i^{[1]} z_i^{[1]} && \text{from } r(N-L) \text{ copies,}\\
&x_i^{[1]} y_0^{[0]} z_i^{[1]} && \text{from } t(N-L) \text{ copies,}\\
&x_i^{[1]} y_i^{[1]} z_0^{[0]} && \text{from } (N-L) \text{ copies,}\\
&x_0^{[0]} y_0^{[0]} z_{q+1}^{[2]} && \text{from } L \text{ copies,}\\
&x_0^{[0]} y_{q+1}^{[2]} z_0^{[0]} && \text{from } tL \text{ copies, and}\\
&x_{q+1}^{[2]} y_0^{[0]} z_0^{[0]} && \text{from } rL \text{ copies.}
\end{aligned}
$$

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

$$\omega(t,1,r) \leqslant \frac{1}{(1-\beta)\log q} \times \log\left(\frac{\left(\begin{array}{l}(t\beta)^{t\beta}((1+r)(1-\beta))^{(1+r)(1-\beta)} \\ \quad \times (t(1-\beta)+(1+r)\beta)^{(t(1-\beta)+(1+r)\beta)}(q+2)^{(t+1+r)}\end{array}\right)}{(t+1+r)^{(t+1+r)}}\right). \tag{7.3}$$

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 of* $\langle n, n, n^r \rangle$ *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 $g(1) = 2.375477...$ 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 $\langle n, n, n^r \rangle$ based on the algorithm of subsection 7.1. By combining the results of Sections 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 $r = 1$, $f(1) = 2.3879$ 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 $\beta = 0.0336$, $f(1.171) = 2.546462806... < g(1.171) = 2.546477....$

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

8.2. *The Case of* $\langle n, n, n^r \rangle$ *with* $0 \leqslant r \leqslant 1$

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 $(0, 2+\varepsilon)$ 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

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

Here is its derivation:

$$\omega(1, 1, r) \leqslant 2 + o(1), \qquad 0 \leqslant r \leqslant 0.294 = \alpha$$

comes from [Co96], and we also have

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

Indeed,

$$\begin{aligned} M(n, n, n^r) &= M(n^{(1-r)/(1-\alpha)} \cdot n^{(r-\alpha)/(1-\alpha)}, n^{(1-r)/(1-\alpha)} \\ &\quad \times n^{(r-\alpha)/(1-\alpha)}, n^{((1-r)\,\alpha)/(1-\alpha)} \cdot n^{(r-\alpha)/(1-\alpha)}) \\ &\leqslant M(n^{(1-r)/(1-\alpha)}, n^{(1-r)/(1-\alpha)}, n^{((1-r)\,\alpha)/(1-\alpha)}) \\ &\quad \times M(n^{(r-\alpha)/(1-\alpha)}, n^{(r-\alpha)/(1-\alpha)}, n^{(r-\alpha)/(1-\alpha)}) \\ &= O((n^{(1-r)/(1-\alpha)})^{2+\varepsilon}\,(n^{(r-\alpha)/(1-\alpha)})^{\omega}) \\ &= O(n^{(2(1-r)+(r-\alpha)\,\omega)/(1-\alpha)}). \end{aligned}$$

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

8.3. *The Case of $\langle n^t, n, n^r \rangle$ 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 $\langle n^t, n, n^r \rangle$. Then*

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

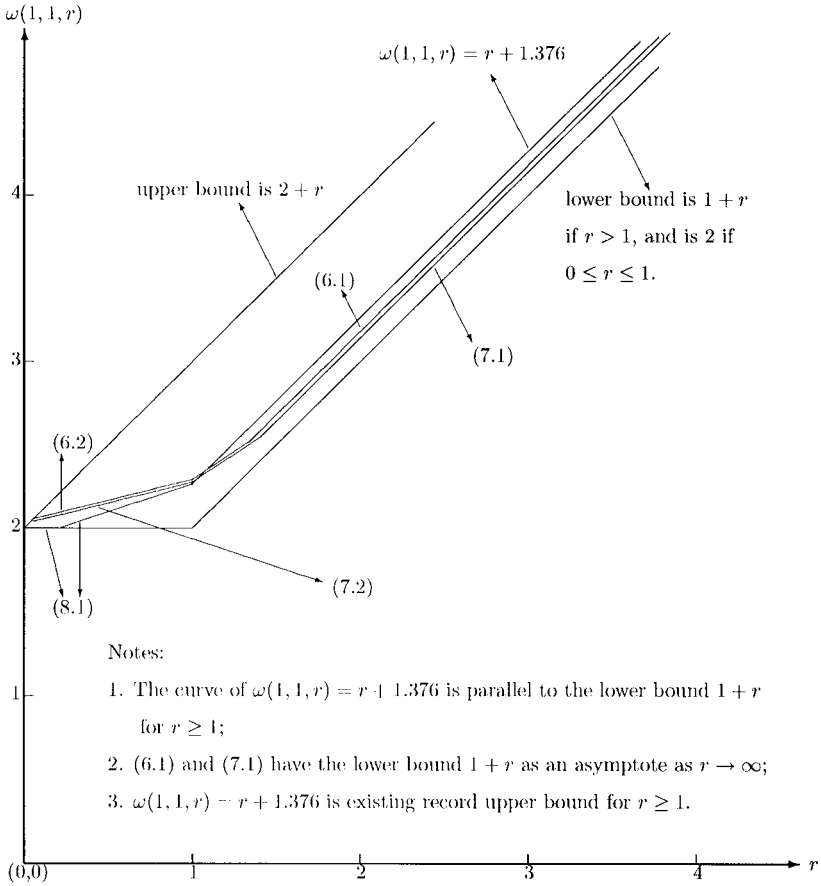

FIG. 1. Illustration of exponent curves $\omega(1, 1, r)$ for $\langle n, n, n^r \rangle$, $0 \leqslant r < +\infty$. (6.1), (6.2), (7.1), (7.2) and (8.1) refer to the respective equations of this paper.

Proof. For $0 \leqslant t \leqslant 0.294 = \alpha$, we have

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

that is, $\omega(t, 1, r) \leqslant r + 1 + \varepsilon$.

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

$$\begin{aligned}
M(n^t, n, n^r) &= M(n^r, n, n^t) \\
&= M(n^{r-(t-\alpha)/(1-\alpha)} \cdot n^{(t-\alpha)/(1-\alpha)}, n^{(1-t)/(1-\alpha)} \\
&\quad \times n^{(t-\alpha)/(1-\alpha)}, n^{((1-t)\alpha)/(1-\alpha)} \cdot n^{(t-\alpha)/(1-\alpha)}) \\
&\leqslant M(n^{r-(t-\alpha)/(1-\alpha)}, n^{(1-t)/(1-\alpha)}, n^{((1-t)\alpha)/(1-\alpha)}) \\
&\quad \times M(n^{(t-\alpha)/(1-\alpha)}, n^{(t-\alpha)/(1-\alpha)}, n^{(t-\alpha)/(1-\alpha)}) \\
&= O((n^{r-(t-\alpha)/(1-\alpha)+(1-t)/(1-\alpha)+\varepsilon})(n^{(t-\alpha)/(1-\alpha)})^{\omega}) \\
&= O(n^{r-(t-\alpha)/(1-\alpha)+(1-t)/(1-\alpha)+(\omega(t-\alpha))/(1-\alpha)+\varepsilon}) \\
&= O(n^{(r(1-\alpha)+(1-t)+(\omega-1)(t-\alpha))/(1-\alpha)+\varepsilon}). \quad \blacksquare
\end{aligned}$$

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

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

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

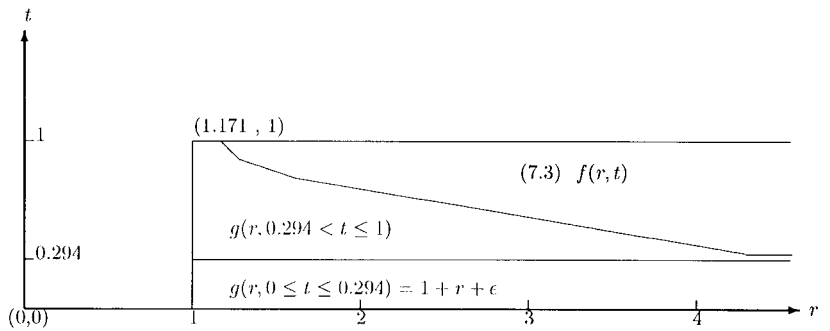

FIG. 2. The three areas are, respectively, the optimal region of the three exponent functions for $\langle n^t, n, n^r \rangle$, $0 \leqslant t \leqslant 1 \leqslant r$.

PART II. APPLICATIONS OF FAST RECTANGULAR MATRIX MULTIPLICATION

9. APPLICATION TO PARALLEL MATRIX COMPUTATIONS

In this section, we will assume the customary EREW PRAM machine model of parallel computing [EG88], [KR90] and apply the results of Section 8 in order to improve the record work-complexity deterministic estimates of [GP89] for fast (NC) 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 9.1. $P(n)$ is the minimum number of arithmetic processors, and $W(n) = O(P(n) \log^2 n)$ is the minimum arithmetic work (that is, the product of time and processor bounds) 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, and

$$W(*, m, n, p) = O(P(n) \log(mnp))$$

is the minimum arithmetic work supporting $O(\log(mnp))$ parallel time bound for multiplication of $m \times n$ by $n \times p$ matrices; $P(*, n) = P(*, n, n, n)$, $W(*, n) = W(*, n, n, n)$.

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

THEOREM 9.1. *The solution to Problems* (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, yielding the work-complexity bound

$$\begin{aligned} W(det, n) &= O(P(det, n) \log^2 n) \\ &= \max\{W(*, n^{1.25}, n, n^{1.25}), W(*, n^{0.5}, n^2, n^{0.5})\}. \end{aligned}$$

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*

$$vu \leqslant n+1 \leqslant (v+1)\,u.$$

This yields the work bound

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

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 simultaneously* $O(\log^2 n)$ *steps,* $P(n) = O(n^{2.851})$ *arithmetic processors and* $W(n) = O(n^{2.851})$ *arithmetic work.*

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* $O(n^{\bar{\omega}(t,\,1,\,r)\,s})$ *arithmetic processors, where* $n > 1$, $s \to \infty$, *and* $\bar{\omega}(t, 1, r)$ *is any number exceeding the value* $\omega(t, 1, r)$ *defined in Section* 2.

Proof. With no loss of generally, 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. 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 $s = 1, 2, \ldots$, and we have

$$t_{s+1} \leqslant t_s + (1 + \max(r, t)) \log_2 n + \log_2 M + 4,$$
$$p_{s+1} \leqslant \max\{n^{(r+t+2)(s+1)}, n^{(r+t)(s+1)} M, p_s M\},$$

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 $M \leqslant n^{\bar{\omega}(t,\,1,\,r)}$ 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 bounds on $P(n)$ and $W(n)$ 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

$$\langle n^{1.25}, n, n^{1.25}\rangle, \qquad \langle n^{1/3}, n^{2/3}, n^2\rangle,$$
$$\langle n, n^{4/3}, n\rangle, \qquad \langle n^{0.5}, n^2, n^{0.5}\rangle,$$

defined by the four following exponents:

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

By applying the results of Section 8, we obtain that

$$\omega(1.25, 1, 1.25) = 1.25\omega(1, 1, 0.8) = 2.8368... < 2.837$$
(by applying (8.1) for $\omega = 2.376$),
$$\omega(1/3, 2/3, 2) = \tfrac{2}{3}\omega(0.5, 1, 3) = 2.7398...$$
(by applying (8.2) for $\omega = 2.376$),
$$\omega(1, 4/3, 1) = \omega(1, 1, 1.33...) = 2.6993...$$
(by selecting $q = 7$, $\beta = 0.033$ in (7.1)),
$$\omega(0.5, 2, 0.5) = 0.5\omega(1, 1, 4) = 2.6390...$$
(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 bounds $W(n) = O(n^{2.837})$ and $P(n) = O(W(n))$.

Remark 9.1. The bound $W(n) = O(n^{2.837})$ 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

$$W(n) = O(\max\{W_1(n), W_2(n), W_3(n), W_4(n)\}),$$

where

$$W_1(n) = n^{\omega_1}, \qquad \omega_1 = \omega(1.25, 1, 1.25)$$
$$= 1.25\,\frac{0.4 + (0.8 - \alpha)\,\omega}{1 - \alpha} \quad [\text{cf. (8.1)}],$$

$$W_2(n) = n^{\omega_2}, \qquad \omega_2 = \omega(1/3, 2/3, 2) = \frac{2}{3}\omega(0.5, 1, 3)$$

$$= \left(\frac{2}{3}\right)\frac{3(1-\alpha) + 0.5 + (\omega - 1)(0.5 - \alpha)}{1-\alpha} \quad [\text{cf. (8.2)}],$$

$$W_3(n) = n^{\omega_3}, \qquad \omega_3 = \omega(1, 4/3, 1) < 2.7,$$

$$W_4(n) = n^{\omega_4}, \qquad \omega_4 = \omega(0.5, 2, 0.5) < 2.64.$$

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

Remark 9.2. Randomized parallel solution of the listed problems (1)–(3) only requires polylogarithmic time and work $O(n^{2.376})$ (cf. [KP91], [KP92], [KP94], [BP94], [P96], [E97]).

10. ACCELERATION OF POLYNOMIAL COMPOSITION AND FACTORIZATION OF POLYNOMIALS OVER FINITE FIELDS

We will extend our results of Part I to accelerate polynomial composition and factorization. To reach the maximum effect, we will modify some of the known reductions of these polynomial computations to matrix multiplication (see subsection 10.5).

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 polynomial composition and the factorization of univariate polynomials over finite fields, which are major problems of algebraic computing. We refer the reader to [GS92] and [KS95] on the background of the latter fundamental problem.

10.2. *Some Definitions and Preliminary Results*

For reader's convenience, in this subsection, we restate some definitions and results of Part I, that we will apply in this section. "Ops" stands for "arithmetic operations," and "bms" stands for "bilinear multiplications." $\langle m, n, p\rangle$ denotes the problem of multiplying a pair of $m \times n$ by $n \times p$ matrices. We represent the complexity of $\langle n^r, n^s, n^t\rangle$ by the number of bilinear multiplications (bms) required, $M(n^r, n^s, n^t)$.

THEOREM 10.1 (Part I, Section 5). *The problem $\langle n, n, n^2\rangle$ of rectangular matrix multiplication can be solved by using $O(n^{3.333953\ldots})$ bms, that is, the exponent of the arithmetic complexity of $\langle n, n, n^2\rangle$ is $\omega(1, 1, 2) \leqslant 3.3333953\ldots$.*

THEOREM 10.2 (Part I, Section 7). *The problem* $\langle n, n, n^r \rangle$ *of rectangular matrix multiplication can be solved by using* $O(n^{\omega(1,1,r)})$ *bms, where* $r \geqslant 1$ *is a rational number, the matrix exponent* $\omega(1, 1, r)$ *is bounded as*

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

where $l \geqslant 2$ *is an integer and* $0 \leqslant b \leqslant 1$.

THEOREM 10.3 (Part I, Section 8). *The problem* $\langle n^t, n, n^r \rangle$ *of Rectangular Matrix Multiplication (where* $0 \leqslant t \leqslant 0.294$, $r \geqslant 1$*) can be solved by using* $O(n^{r+1+\varepsilon})$ *bms.*

10.3. *Complexity of Modular Polynomial Composition*

THEOREM 10.4. *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. The arithmetic complexity of computing the coefficients of the polynomial

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

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

Proof. Algorithm 2.1 of [BK78] for computing $p(q(x)) \bmod x^{n+1}$ has its complexity dominated by the complexity of the problem $\langle n, \sqrt{n}, \sqrt{n} \rangle$. Consequently, Theorem 10.4 immediately follows from Theorem 10.1. ∎

For comparison, the known exponent for the complexity of the above problem of modular polynomial composition was obtained by reduction of the problem $\langle n, \sqrt{n}, \sqrt{n} \rangle$ to $\sqrt{n}$ blocks of square $\sqrt{n} \times \sqrt{n}$ matrix multiplication, so that

$$M(n, \sqrt{n}, \sqrt{n}) \leqslant \sqrt{n}\, M(\sqrt{n}, \sqrt{n}, \sqrt{n})$$

(cf. e.g., [KS95]). The resulting exponent 1.688 exceeds one of Theorem 10.4 by 0.021.

10.4. *Factorization of Polynomial over Finite Fields (Two Approaches)*

There are two major approaches to the factorization of a univariate polynomial of a degree n over the finite field $\mathbf{F}_q$ with q elements. These approaches are due to Berlekamp [B70] and Cantor and Zassenhaus [CZ81]. Both of the approaches lead to randomized algorithms and were recently improved in [GS92] and [KS95], to yield the current record complexity estimates for the factorization problem. We will show further improvement of all these record estimates, by using fast matrix multiplication. We will follow the flowchart of [KS95], where the two cited approaches are treated separately and the Cantor/Zassenhaus approach is partitioned into the study of the two cases, of the equal-degree and the distinct-degree factorization of a polynomial, where all the factors must have the same (equal) degree or all must have distinct degrees, respectively. We will study these two approaches (one with two subcases) in the next three sections.

Our Theorems 10.5–10.7 will specify our record complexity estimates for polynomial factorization, which depend on the two parameters, n (the degree of the polynomial) and q (the cardinality of the field). In particular, we yield the factorization over $\mathbf{F}_q$ by using $O(n^{1.8356}+n^{1.763}\log q)$ field operations or alternatively, $O(n^{1.80535}\log q)$, versus the previous record bounds $O(n^{1.852}+n^{1.763}\log q)$ and $O(n^{1.815}\log q)$ of [KS95]. As in [KS95], our latter record bound is obtained based on each of the two approaches, that is, Cantor/Zassenhaus' and Berlekamp's.

10.5. *Complexity of Equal-Degree Polynomial Factorization over Finite Fields*

The probabilistic algorithm of von zur Gathen and Shoup (cf. [GS92]) solves the equal-degree factorization problem for a univariate polynomial of a degree n over the finite field $\mathbf{F}_q$ with q elements by 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 the estimated complexity of polynomial composition modulo x^n. Due to our Theorems 10.1 and 10.4 and Remark 2.1, the bound on the complexity of the equal-degree factorization problem can be immediately improved as follows.

THEOREM 10.5. *The equal-degree factorization of a univariate polynomial of a degree n over the finite field* $\mathbf{F}_q$ *with q elements an be computed probabilistically by using an expected number of*

$$O(M(n, \sqrt{n}, \sqrt{n})) = O(n^{1.667} + n^{1+o(1)} \log q)$$

operations in $\mathbf{F}_q$.

10.6. *Complexity of Distinct-Degree Factorization over a Finite Field*

Section 2 of [KS95] presents a (deterministic) algorithm (Algorithm D) for the distinct-degree factorization of a polynomial of a degree n over the finite field $\mathbf{F}_q$ with q elements. The algorithm 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 $0 \leqslant \beta \leqslant 1$ (see Theorem 3 in [KS95]).

By substituting $\omega < 2.375477$ of [CW90] and then minimizing the exponent of n, Kaltofen and Shoup obtained the estimate of $O(n^{1.815} \log q)$ operations in $\mathbf{F}_q$ (cf. [KS95], Theorem 3).

We will next improve this bound as follows:

THEOREM 10.6. *Distinct degree factorization of a univariate polynomial over a finite field* $\mathbf{F}_q$ *with q elements can be computed deterministically by using*

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

operations in $\mathbf{F}_q$, *for any* β *from the interval* $0 \leqslant \beta \leqslant 1$. *For* $\beta = 0.805347$, *this bound can be turned into* $O(n^{1.80535} \log q)$.

By comparing Theorems 10.5 and 10.6, we conclude that the estimate of Theorem 10.6 is larger and dominates the overall asymptotic complexity of polynomial factorization over $\mathbf{F}_q$ in terms of the number of the field operations used, although we need randomization to apply the estimates of Theorem 10.5 and do not need it to apply Theorem 10.6.

Proof. To prove Theorem 10.6, we will first recall and improve Lemmas 3 and 4 of [KS95].

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* $g_1, g_2, ..., g_k, h \in \mathbf{K}[x]$, *all of degrees less than n, where* $k = O(n^{\delta})$, $0 \leqslant \delta \leqslant 1$, *it suffices to apply*

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

operations in $\mathbf{K}$ *to compute*

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

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

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

We will 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 algorithms for rectangular matrix multiplication achieve better results than application of 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* $O(n^{(\omega+1)/2} K^{(\omega-1)/2})$ *operations in* $\mathbf{F}_q$ *suffices to compute*

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

for $K = O(n^\delta)$, $0 \leqslant \delta \leqslant 1$.

For the sake of completeness of our argument, let us outline the short proof of Lemma 4.

Proof of Lemma 4 *of* [KS95]. For $i \geqslant 1$, let $G_i = x^{q^{ir}} \bmod f \in \mathbf{F}_q[x]$. Assume that we have computed $G_1, \ldots, G_m$. Then we can obtain $G_{m+1}, \ldots, G_{2m}$ by computing

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

by means of the algorithm supporting Lemma 3. Therefore, to compute $G_1, \ldots, G_K$ 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 specified in the following way where we incorporate our improved 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 $\langle n, \sqrt{n}, \sqrt{n} \rangle$ (i.e., let $k = 1$ in Lemma 3).

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

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

is equivalent to solving the problem $\langle n, \sqrt{2n}, \sqrt{2n}\rangle$ (i.e., let $k=2$ in Lemma 3).

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

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

is equivalent to solving the problem $\langle n, \sqrt{n(K/4)}, \sqrt{n(K/4)}\rangle$ (i.e., let $k=K/4$ in Lemma 3).

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

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

is equivalent to solving the problem $\langle n, \sqrt{n(K/2)}, \sqrt{n(K/2)}\rangle$, (i.e., let $k=K/2$ in Lemma 3).

We recall that

$$M(n, \sqrt{n(K/2^{i+1})}, \sqrt{n(K/2^{i+1})}) \leqslant \tfrac{1}{2} M(n, \sqrt{n(K/2^i)}, \sqrt{n(K/2^i)}), \qquad i=1, 2, ..., \log K.$$

Now we sum the complexity estimates for all steps from 1 to $\log K$, to arrive at the overall complexity bound of

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

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

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

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

The latter argument enables us to replace the term

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

in the estimate of Theorem 3 of [KS95] by

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

so as to yield the bound of

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

on the complexity of the distinct-degree factorization in $\mathbf{F}_q$ (cf. our Remark 2.1). To minimize the exponent of n in the latter bound, we choose $\beta = 0.805347$. Furthermore, in Theorem 10.2 of our subsection 10.2, we choose $b = 0.023$ and $l = 8$. Then we arrive at the estimate

$$\omega(1,\,1-\beta/2,\,1-\beta/2) \leqslant 1.805346859... < 1.80535.$$

Since $\beta + o(1)$ is bounded from above by 0.80535, we finally arrive at the complexity bound $O(n^{1.80535} \log q)$, thus completing the proof of Theorem 10.6 to yield a new record complexity estimate for the distinct-degree factorization (and consequently, for the entire factorization algorithm). ▮

10.7. *Complexity of the Fast Black Box Berlekamp Algorithm*

In this subsection, we will follow the line of Section 3 of [KS95] but will utilize the results of our Part I on rectangular matrix multiplication to improve the estimates of [KS95] for the complexity of the fast randomized Black Box Berlekamp Algorithm. The latter algorithm is a version of Berlekamp's algorithm ameliorated in [KS95] for the factorization of a monic square-free polynomial over the finite field $\mathbf{F}_q$ with q elements. By following [KS95], we will refer to this algorithm as *Algorithm B*.

First, let us recall the result of Theorem 4 of [KS95], which states that *for any constant* β *with* $0 \leqslant \beta \leqslant 1$, *Algorithm B of* [*KS*95] *can be implemented so as to use an expected number of*

$$O(n^{(\omega+1)/2+(3-\omega)\,|\beta-1/2|+o(1)} + n^{(\omega+1)/2+(1-\beta)+o(1)} + n^{1+\beta+o(1)} \log q) \tag{10.1}$$

operations in $\mathbf{F}_q$. *By choosing* $\omega < 2.375477$ *and minimizing the exponent* n, *one obtains the bound of*

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

operations in $\mathbf{F}_q$.

We will next improve the latter bound to

$$O(\min\{n^{1.860} + n^{1.808} \log q,\, n^{1.8335} \log q\}\,);$$

then we will also improve a refined estimate of [KS95] for the complexity of Algorithm B. Note first that the term $O(n^{1.880}+n^{1.808}\log q)$ is obtained by choosing $\beta=0.808$ in (10.1); also note that $n^{(\omega+1)/2}$ comes from the complexity bound for the problem of modular polynomial composition, which we bound by $O(n^{3.334/2})=O(n^{1.667})$, due to our Theorems 10.1 and 10.4. Then, we will bound the exponents of the first and the second terms by 1.8591... and of the third term by 1.808, that is, we have the overall complexity bound of $O(n^{1.860}+n^{1.808}\log q)$.

To yield the estimate $O(n^{1.8335}\log q)$, we first note that $(\omega+1)/2$ in the second term of (10.1) can be replaced by $\omega(1,1,2)<3.333953$, then optimize the exponent of n by choosing an appropriate β, to bound the sum by $O(n^{1.8335}\log q)$, and then, finally, prove that the exponent of the first term can also be decreased to 1.8335. Towards the latter goal, 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 $\langle m, t, n\rangle$, 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 $(1-\beta)/\beta<0.294$. By applying our Theorem 10.3, we obtain

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

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

Theorem 4 is also improved by Theorem 5 of [KS95], which gives us the record randomized complexity estimate for factorization over $\mathbf{F}_q$ by the Fast Black Box Berlekamp Algorithm. Our results of Part I will enable us to improve the estimates of Theorem 5 of [KS95] too.

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

$$O(n^{(\omega+1)/2+(3-\omega)|\beta-1/2|+o(1)} + n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)} + n^{1+\beta+o(1)} \log q) \tag{10.2}$$

operations in $\mathbf{F}_q$. In particular, by choosing $\omega < 2.375477$ and minimizing the exponent of n, one obtains 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 (10.2) *can be reduced to*

$$O(n^{(\omega+1)/2+(1-\beta)(\omega-1)/2+o(1)} + n^{1+\beta+o(1)} \log q) \tag{10.3}$$

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

We will next prove the record estimates for the asymptotic complexity of polynomial factorization over $\mathbf{F}_q$ based on Algorithm B.

THEOREM 10.7. *The bounds $O(n^{1.852} + n^{1.763} \log q)$ and $O(n^{1.815} \log q)$ on the complexity of the nth degree polynomial factorization over $\mathbf{F}_q$ based on Algorithm B of* [KS95] *can be improved to yield the bounds $O(n^{1.8356} + n^{1.763} \log q)$ and $O(n^{1.80535} \log q)$, respectively.*

Proof. Note that the second term of (10.2),

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

comes from the solution of the problem $\langle n, n^{1-\beta/2}, n^{1-\beta/2} \rangle$ (cf. our Theorem 10.4) and thus can be replaced by

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

(as in the preceding section where we improved the result of Theorem 3 of [KS95]). By choosing again $b = 0.023$ and $l = 8$ in Theorem 10.2 of subsection 10.2 (as we did at the end of subsection 10.6) but choosing $\beta = 0.763 - o(1)$, we arrive at the bound

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

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 $M(n^{1-\beta}, n, n^{\beta})$. For $\beta = 0.763 - o(1)$, we have

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

On the other hand,

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

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

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

Therefore, the expression

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

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

Finally, we discuss the improvement from $O(n^{1.1815} \log q)$ to $O(n^{1.80535} \log q)$. Since the second and the third terms of Theorem 5 of [KS95] [cf. (10.2)] are the same as in the Theorem 3 of [KS95], that is, bounded by $O(n^{1.80535} \log q)$, it remains to prove that the first term is dominated by $O(n^{1.80535})$. Choose $\beta = 0.80535 - o(1)$ and note that (as we mentioned above) the first term comes from the bound

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

Now, due to the inequality

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

and the application of our Theorem 10.3, we arrive at the desired bound of $O(n^{1.80535} \log q)$, thus proving Theorem 10.7. ∎

11. APPLICATION TO FINDING BASIC SOLUTIONS OF A LINEAR PROGRAMMING PROBLEM

In this section, we will apply the results on rectangular matrix multiplications from Section 8 to find basic solutions of a linear programming problem with m constraints and n variables and improve its record complexity estimate from $O(n^{1.594})$ to $O(m^{1.575})$.

First, let us follow [BM98] and briefly describe the problem.

PROBLEM 11.1 (Basis Crashing for a Linear Programming Problem). *Consider the standard-form system of linear constraints*

$$Ax = b, \qquad x \geqslant 0,$$

where $A \in \mathbf{R}^{m \times n}$ *is assumed to have m linearly independent rows,* $b \in \mathbf{R}^m$, *and* $x \in \mathbf{R}^n$, $\mathbf{R}$ *denoting the field of real numbers. A solution x of this system is said to be* basic *if the set of columns* $A_{\cdot j}$ *with* $x_j \neq 0$ *is linearly independent. Thus, a basic solution has at most m positive components.*

The problem of finding a basic solution given a non-basic one arises frequently in linear programming, especially in the context of interior-point methods. For simplicity, we call this problem *basis crashing*.

P. A. Belling and N. Megiddo in [BM98] reduced this problem to performing rectangular matrix multiplication and proved the following estimate.

THEOREM 11.1. *Problem* 11.1 *can be solved by using*

$$O((m^{1+2t} + m^{\omega(1,1,t)})\, n/m^t) = O((m^{1+t} + m^{\omega(1,1,t)-t})\, n) \qquad (11.1)$$

arithmetic operations, for any t in the interval $0 \leqslant t \leqslant 1$.

To minimize (11.1), we seek t, $0 \leqslant t \leqslant 1$, which minimizes

$$\max\{1 + t, \omega(1, 1, t) - t\}.$$

Substitute (8.1) and rewrite this expression as

$$\max\left\{1 + t, \frac{2(1-t) + (t-\alpha)\,\omega}{1-\alpha} - t\right\}, \qquad (11.2)$$

where $\omega = 2.376$ and $\alpha = 0.294$. The minimum is reached for $t = 0.5747\ldots < 0.575$, which implies the following estimate.

THEOREM 11.2. *The complexity of Problem* 11.1 *of computing basis solutions to the linear programming problem with m constraints and n variables is* $O(m^{1.575}n)$.

Remark 11.1. In [BM98] the estimate similar to (11.2) was obtained, but for some reason, a slightly larger numerical value of the exponent of m (namely, 1.594) was deduced.

ACKNOWLEDGMENT

Erich Kaltofen pointed out to us the application of rectangular matrix multiplication to polynomial factorization.

REFERENCES

[B70] Berlekamp, E. R. (1970), Factoring polynomials over large finite fields, *Math. Comput.* **24**, 713–735.

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

[BCS97] Bürgisser, P., Clausen, M., and Shokrollahi, M. A. (1997), "Algebraic Computational Complexity," Springer, Berlin.

[BD76] Brockett, R. W., and Dobkin, D. (1976), On the number of multiplications required for matrix multiplications, *SIAM J. Complexity* **5**, 624–628.

[Be46] Behrend, F. A. (1946), On sets of integers which contain no three terms in arithmetical progression, *Proc. Nat. Acad. Sci. USA* **32**, 331–332.

[BK78] Brent, R. P., and Kung, H. T. (1978), Fast algorithms for manipulating formal power series, *J. Assoc. Comput. Mach.* **25**, 581–595.

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

[BM98] Beling, P. A., and Megiddo, N. (1998), Using fast matrix multiplication to find basic solutions, *Theoret. Comput. Sci.* to appear.

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

[Co82] Coppersmith, D. (1982), Rapid multiplication of rectangular matrices, *SIAM J. Comput.* **11**, 467–471.

[Co97] Coppersmith, D. (1997), Rectangular matrix multiplication revisited, *J. Complexity* **13**, 42–49.

[CW82] Coppersmith, D., and Winograd, S. (1982), On the asymptotic complexity of matrix multiplication, *SIAM J. Comput.* **11**, 472–492.

[CW90] Coppersmith, D., and Winograd, S. (1990), Matrix multiplication via arithmetic progressions, *J. Symbolic Comput.* **9**, 251–280.

[CZ81] Cantor, D. G., and Zassenhaus, H. (1981), A new algorithm for factoring polynomials over finite fields, *Math. Comput.* **36**, 587–592.

[E97] Eberly, W. (1997), Parallel matrix inversion over abstract fields: Two approaches, *in* "Proceedings Second International Symposium on Parallel Symbolic Computation (PASCO'97)," pp. 38–45, ACM Press, New York.

[EG88] Eppstein, D., and Galil, Z. (1988), Parallel algorithmic techniques for combinatorial computation, *Annual Rev. Comput. Sci.* **3**, 233–283.

[GL96] Golub, G. H., and Van Loan, C. F. (1996), "Matrix Computations," 3th ed., Johns Hopkins Univ. Press, Baltimore.

[GP89] Galil, Z., and Pan, V. Y. (1989), Parallel evaluation of the determinant and of the inverse of a matrix, *Infor. Proc. Lett.* **30**, 41–45.

[GS92] von zur Gathen, J., and Shoup, V. (1992), Computing Frobenius maps and factoring polynomials, *Comput. Complexity* **2**, 187–224.

[KP91] Kaltofen, E., and Pan, V. Y. (1991), Processor efficient parallel solution of linear systems over an abstract field, *in* "Proceedings, 3rd Annual ACM Symposium on Parallel Algorithms and Architectures," pp. 180–191, ACM Press, New York.

[KP92] Kaltofen, E., and Pan, V. Y. (1992), Processor-efficient parallel solution of linear systems. II. The positive characteristic and singular case, *in* "Proceedings of 33rd Annual IEEE Symposium on Foundations of Computer Science," pp. 714–723, IEEE Computer Society Press, Los Alamitos, CA.

[KP94] Kaltofen, E., and Pan, V. Y. (1994), Parallel solution of Toeplitz and Toeplitz-like linear systems over fields of small positive characteristic, *in* "Proceedings of First International Symposium on Parallel Symbolic Computation (PASCO'94), Linz, Austria (Sept. 1994)," Lecture Notes Series in Computing, Vol. 5, pp. 225–233, World Scientific, Singapore.

[KR90] Karp, R., and Ramachandran, V. (1990), A survey of parallel algorithms for shared memory machines, *in* "Handbook for Theoretical Computer Science" (J. van Leeuwen, Ed.), pp. 869–941, North Holland, Amsterdam.

[KS95] Kaltofen, E., and Shoup, V. (1995), Subquadratic-time factoring of polynomials over finite fields, *in* "Proceedings, 27th Annual ACM Symposium on Theory Comput.," pp. 398–406, ACM Press, New York; *Math. Comput.*, in press.

[Pan72] Pan, V. Y. (1972), On schemes for the computation of products and inverses of matrices, *Uspekhi Mat. Nauk.* **27**, 249–250. [in Russian]

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

[Pan,a] Pan, V. Y. (1984), How can we speed-up matrix multiplication?, *SIAM Rev.* **26**, 393–415.

[P96] Pan, V. Y. (1996), Parallel computation of polynomial GCD and some related parallel computations over abstract fields, *Theor. Comput. Sci.* **162**, 173–233.

[Sc81] Schönhage, A. (1981), Partial and total matrix multiplication, *SIAM J. Comput.* **10**, 434–456.

[SS42] Salem, R., and Spencer, D. C. (1942), On sets of integers which contain no three terms in arithmetical progression, *Proc. Nat. Acad. Sci. USA* **28**, 561–563.

[St69] Strassen, V. (1969), Gaussian elimination is not optimal, *Numerische Math.* **13**, 354–356.

[St86] Strassen, V. (1986), The asymptotic spectrum of tensors and the exponent of matrix multiplication, *in* "Proceedings, 27th Annual IEEE Symposium on Foundations of Computer Science," pp. 49–54, IEEE Computer Society Press.

[St87] Strassen, V. (1987), Relative bilinear complexity and matrix multiplication, *J. Reine Angew. Math.* **375/376**, 406–443.

[St88] Strassen, V. (1988), The asymptotic spectrum of tensor, *J. Reine Angew. Math.* **384**, 102–152.

Printed in Belgium