Faster Algorithms for Rectangular Matrix Multiplication

François Le Gall
Department of Computer Science
The University of Tokyo
Tokyo, Japan
e-mail: legall@is.s.u-tokyo.ac.jp

***Abstract*—Let α be the maximal value such that the product of an $n \times n^{\alpha}$ matrix by an $n^{\alpha} \times n$ matrix can be computed with $n^{2+o(1)}$ arithmetic operations. In this paper we show that $\alpha > 0.30298$, which improves the previous record $\alpha > 0.29462$ by Coppersmith (Journal of Complexity, 1997). More generally, we construct a new algorithm for multiplying an $n \times n^k$ matrix by an $n^k \times n$ matrix, for any value $k \neq 1$. The complexity of this algorithm is better than all known algorithms for rectangular matrix multiplication. In the case of square matrix multiplication (i.e., for $k = 1$), we recover exactly the complexity of the algorithm by Coppersmith and Winograd (Journal of Symbolic Computation, 1990).**

These new upper bounds can be used to improve the time complexity of several known algorithms that rely on rectangular matrix multiplication. For example, we directly obtain a $O(n^{2.5302})$-time algorithm for the all-pairs shortest paths problem over directed graphs with small integer weights, where n denotes the number of vertices, and also improve the time complexity of sparse square matrix multiplication.

***Keywords*-matrix multiplication; rectangular matrices; algorithms;**

I. Introduction

Matrix multiplication is one of the most fundamental problems in computer science and mathematics. Besides the fact that several computational problems in linear algebra can be reduced to the computation of the product of two matrices, the complexity of matrix multiplication also arises as a bottleneck in a multitude of other computational tasks (e.g., graph algorithms). The standard method for multiplying two $n \times n$ matrices uses $O(n^3)$ arithmetic operations. Strassen showed in 1969 that this trivial algorithm is not optimal, and gave a algorithm that uses only $O(n^{2.808})$ arithmetic operations. This has been the beginning of a long story of improvements that lead to the upper bound $O(n^{2.376})$ by Coppersmith and Winograd [9], which has been recently further improved to $O(n^{2.3727})$ by Vassilevska Williams [23]. A slightly weaker improvement has also been found by Stothers [21]. Note that all the above complexities refer to the number of arithmetic operations involved, but naturally the same upper bounds hold for the time complexity as well when each arithmetic operation can be done in negligible time (e.g., in $\mathrm{poly}(\log n)$ time).

Finding the optimal value of the exponent of square matrix multiplication is naturally one of the most important open problems in algebraic complexity. It is widely believed that the product of two $n \times n$ matrices can be computed with $O(n^{2+\epsilon})$ arithmetic operations for any constant $\epsilon > 0$. Several conjectures, including conjectures about combinatorial structures [9] and about group theory [6], [5], would, if true, lead to this result (see also [1] for recent work on these conjectures). Another way to interpret this open problem is by considering the multiplication of an $n \times m$ matrix by an $m \times n$ matrix. Suppose that the matrices are defined over a field. For any $k > 0$, define the exponent of such a rectangular matrix multiplication as follows:

$$\omega(1,1,k) = \inf\{\tau \in \mathbb{R} \mid C(n,n,\lfloor n^k \rfloor) = O(n^{\tau})\},$$

where $C(n,n,\lfloor n^k \rfloor)$ denotes the minimum number of arithmetic operations needed to multiply an $n \times \lfloor n^k \rfloor$ matrix by an $\lfloor n^k \rfloor \times n$ matrix. Note that, while the value $\omega(1,1,k)$ may depend on the field under consideration, it is known that it can depend only on the characteristic of the field [20]. Define $\omega = \omega(1,1,1)$ and $\alpha = \sup\{k | \omega(1,1,k) = 2\}$. The value ω represents the exponent of square matrix multiplication, and the value α essentially represents the largest value such that the product of an $n \times n^{\alpha}$ matrix by an $n^{\alpha} \times n$ matrix can be computed with $O(n^{2+\epsilon})$ arithmetic operations for any constant ϵ. Since $\omega = 2$ if and only if $\alpha = 1$, one possible strategy towards showing that $\omega = 2$ is to give lower bounds on α. Coppersmith [7] showed in 1982 that $\alpha > 0.172$. Then, based on the techniques developed in [9], Coppersmith [8] improved this lower bound to $\alpha > 0.29462$. This is the best lower bound on α known so far.

Except for Coppersmith's work on the value α, there have been relatively few algorithms that focus specifically on rectangular matrix multiplication. Since it is well known (see, e.g, [16]) that multiplying an $n \times n$ matrix by an $n \times m$ matrix, or an $m \times n$ matrix by an $n \times n$ matrix, can be done with the same number of arithmetic operations as multiplying an $n \times m$ matrix by an $m \times n$ matrix, the value $\omega(1,1,k)$ represents the exponent of all these three types of rectangular matrix multiplications. Note that, by decomposing the product into smaller matrix products, it is easy to obtain (see, e.g, [16]) the following upper bound:

$$\omega(1,1,k) = \begin{cases} 2 & \text{if } 0 \leq k \leq \alpha \\ 2 + (\omega - 2)\frac{k-\alpha}{1-\alpha} & \text{if } \alpha \leq k \leq 1. \end{cases} \quad (1)$$

Lotti and Romani [16] obtained nontrivial upper bounds on

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.80

$\omega(1,1,k)$ based on the seminal result by Coppersmith [7] and on early works on square matrix multiplication. Huang and Pan [12] showed how to apply ideas from [9] to the rectangular setting and obtained the upper bound $\omega(1,1,2) < 3.333954$, but this approach did not lead to any upper bound better than (1) for $k \leq 1$. Ke, Zeng, Han and Pan [15] further improved Huang and Pan's result to $\omega(1,1,2) < 3.2699$, by using again the approach from [9], and also reported the upper bounds $\omega(1,1,0.8) < 2.2356$ and $\omega(1,1,0.5356) < 2.0712$, which are better than those obtained by (1). Their approach, nevertheless, did not give any improvement for the value of α.

The results [8], [12], [15], [21], [23] are all obtained by extending the approach by Coppersmith and Winograd [9]. Informally, the idea is to start with a basic construction (some small trilinear form), and then exploit general properties of matrix multiplication to derive an upper bound on the exponent ω from this construction. The main contributions of [9] consist of two parts: the discovery of new basic constructions and the introduction of strong techniques to analyze them. In their paper, Coppersmith and Winograd actually present three algorithms, based on three different basic constructions. The first basic construction (Section 6 in [9]) is the simplest of the three and leads to the upper bound $\omega < 2.40364$. The second basic construction (Section 7 in [9]), that we will refer in this paper as F_q (here $q \in \mathbb{N}$ is a parameter), leads to the upper bound $\omega < 2.38719$. The third basic construction (Section 8 in [9]) is $F_q \otimes F_q$, the tensor product of two instances of F_q, and leads to the improved upper bound $\omega < 2.375477$. The algorithms for rectangular matrix multiplication [8], [12], [15] already mentioned use a similar approach. Huang and Pan [12] obtained their improvement on $\omega(1,1,2)$ by taking the easiest of the three constructions in [9] and carefully modifying the analysis to evaluate the complexity of rectangular matrix multiplication. Ke, Zeng, Han and Pan [15] obtained their improvements similarly, but by using the second basic construction from [9] (the construction F_q) instead, which lead to better upper bounds. In order to obtain the lower bound $\alpha > 0.29462$, Coppersmith [8] relied on a more complex approach: the basic construction considered is still F_q, but several instances for distinct values of q are combined together in a subtle way in order to keep the complexity of the resulting algorithm small enough (i.e., not larger than $n^{2+o(1)}$).

Besides the fact that a better understanding of $\omega(1,1,k)$ gives insights into the nature of matrix multiplication and ultimately may help showing that $\omega = 2$, fast algorithms for multiplying an $n \times n^k$ matrix by an $n^k \times n$ with $k \neq 1$ have also a multitude of applications. Typical examples not directly related to linear algebra include the construction of fast algorithms for the all-pairs shortest paths problem [2], [18], [25], [28], [29], the dynamic computation of the transitive closure [11], [19], finding ancestors [10], detecting directed cycles [26]. Rectangular matrix multiplication has also been used in computational complexity [17], [24], and to speed-up sparse square matrix multiplication [3], [14], [27] or tasks in computational geometry [13], [14]. Obtaining better upper bounds on $\omega(1,1,k)$ would thus reduce the asymptotic time complexity of algorithms in a wide range of areas. We nevertheless stress that such improvements are only of theoretical interest, since the huge constants involved in the complexity of fast matrix multiplication usually make these algorithms impractical.

A. Statement of our results

In this paper we construct new algorithms for rectangular matrix multiplication, by taking the tensor power $F_q \otimes F_q$ as basic construction and analyzing this construction in the framework of rectangular matrix multiplication. We use these ideas to prove that $\omega(1,1,k) = 2$ for any $k \leq 0.30298$, as stated in the following theorem.

Theorem 1. *For any value $k \leq 0.30298$, the product of an $n \times n^k$ matrix by an $n^k \times n$ matrix can be computed with $O(n^{2+\epsilon})$ arithmetic operations for any constant $\epsilon > 0$.*

More generally, we present an algorithm for multiplying an $n \times n^k$ matrix by an $n^k \times n$ matrix, for any value k. We show that the complexity of this algorithm can be expressed as a (nonlinear) optimization problem, and use this formulation to derive upper bounds on $\omega(1,1,k)$. Table I shows the bounds we obtain for several values of k. The bounds obtained for $0 \leq k \leq 1$ are represented in Figure 1 as well.

k	upper bound on $\omega(1,1,k)$
0.30298	2
0.40	2.012175
0.50	2.046681
0.5302	2.060396
0.55	2.070063
0.60	2.096571
0.70	2.156959
0.80	2.224790
0.90	2.298048
1.00	2.375477
1.25	2.581815
1.50	2.800116
1.75	3.025906
2.00	3.256689
2.50	3.727808
3.00	4.207372
3.50	4.693151
4.00	5.180715
4.50	5.672001
5.00	6.166736

Table I
OUR UPPER BOUNDS ON THE EXPONENT OF THE MULTIPLICATION OF AN $n \times n^k$ MATRIX BY AN $n^k \times n$ MATRIX.

The results of this paper can be seen as a generalization of Coppersmith-Winograd's approach to the rectangular setting. In the case of square matrix multiplication (i.e., for $k = 1$), we recover naturally the same upper bound $\omega(1,1,1) < 2.375477$ as the one obtained in [9]. Let us mention that we can, in a rather straightforward way, combine our results with the upper bound $\omega < 2.3727$ by Vassilevska Williams [23] to obtain slightly improved bounds for $k \approx 1$. The idea is, very similarly to how Equation (1) was obtained, to exploit the convexity of the function $\omega(1,1,k)$. Concretely, for any fixed value $0 \leq k_0 < 1$, the

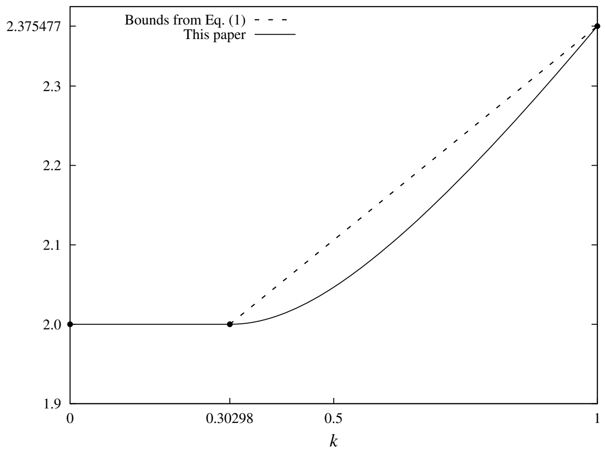

Figure 1. Our upper bounds (in plain line) on $\omega(1,1,k)$, for $0 \leq k \leq 1$. The dashed line represents the upper bounds on $\omega(1,1,k)$ obtained by using Equation (1) with the values $\alpha > 0.30298$ and $\omega < 2.375477$.

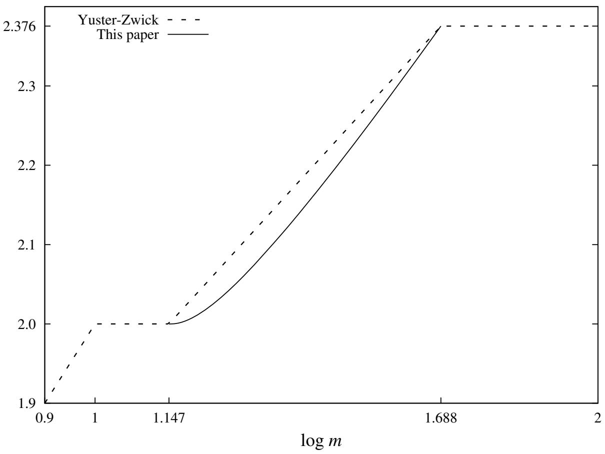

Figure 2. Upper bounds on the exponent for the multiplication two $n \times n$ matrices with at most m non-zero entries. The horizontal axis represents $\log_n(m)$. The dashed line represents the results by Yuster and Zwick [27] and shows that the term $n^{\omega(1,1,\lambda_m)}$ dominates the complexity when $1 \leq \log_n(m) \leq (1+\omega)/2$. The plain line represents our improvements.

inequality

$$\omega(1,1,k) \leq \omega(1,1,k_0) + (\omega - \omega(1,1,k_0))\frac{k-k_0}{1-k_0}$$

holds for any k such that $k_0 \leq k \leq 1$. This enables us to combine an upper bound on $\omega(1,1,k_0)$, for instance one of the values in Table I, with the improved upper bound $\omega < 2.3727$ by Vassilevska Williams. Since the improvement is small and concerns only the case $k \approx 1$, we will not discuss it further.

For $k > 0.29462$ and $k \neq 1$, the complexity of our algorithms is better than all known algorithms for rectangular matrix multiplication, including the algorithms [12], [15] mentioned above. Moreover, for $0.30298 < k < 1$, our new bounds are significantly better than what can be obtained solely from the bound $\alpha > 0.30298$ and $\omega < 2.375477$ through Equation (1), as illustrated in Figure 1. This suggests that non-negligible improvements can be obtained for all applications of rectangular matrix multiplications that rely on this simple linear interpolation, as we discuss below.

B. Applications

In this subsection we describe quantitatively the improvements that our new upper bounds imply for some applications: sparse square matrix multiplication and the all-pairs shortest paths problem.

Sparse square matrix multiplication: Yuster and Zwick [27] have shown how fast algorithms for rectangular matrix multiplication can be used to construct fast algorithms for computing the product of two sparse square matrices (this result has been generalized to the product of sparse rectangular matrices in [14], and the case where the output matrix is also sparse has been studied in [3]). More precisely, let M and M' be two $n \times n$ matrices such that each matrix has at most m non-zero entries, where $0 \leq m \leq n^2$. Yuster and Zwick [27] showed that the product of M and M' can be computed in time

$$O\left(\min(nm, n^{\omega(1,1,\lambda_m)+o(1)}, n^{\omega+o(1)})\right),$$

where λ_m is the solution of the equation $\lambda_m + \omega(1,1,\lambda_m) = 2\log_n(m)$. Using the upper bounds on $\omega(1,1,k)$ of Equation (1) with the values $\alpha < 0.294$ and $\omega < 2.376$, this gives the complexity depicted in Figure 2.

These upper bounds can be of course directly improved by using the new upper bound on ω by Vassilevska Williams [23] and the new lower bound on α given in the present work, but the improvement is small. A more significant improvement can be obtained by using directly the upper bounds on $\omega(1,1,k)$ presented in Figure 1, which gives the new upper bounds on the complexity of sparse matrix multiplication depicted in Figure 2.

The all-pairs shortest paths problem: Zwick [29] has shown how to use rectangular matrix multiplication to compute the all-pairs shortest paths in weighted direct graphs where the weights are bounded integers. The time complexity obtained is $O(n^{2+\mu+\epsilon})$, for any constant $\epsilon > 0$, where μ is the solution of the equation $\omega(1,1,\mu) = 1+2\mu$. Using the upper bounds on $\omega(1,1,k)$ of Equation (1) with $\alpha > 0.294$ and $\omega < 2.376$, this gives $\mu < 0.575$ and thus complexity $O(n^{2.575})$. Actually, this complexity can be further reduced to $O(n^{2.5356})$ using the bounds on $\omega(1,1,k)$ given in [15].

Our results (see Table I) show that $\omega(1,1,0.5302) < 2.0604$, which gives the upper bound $\mu < 0.5302$. We thus obtain the following result.

Theorem 2. *There exists an algorithm that computes the shortest paths between all pairs of vertices in a weighted directed graph with bounded integer weights in* *time $O(n^{2.5302})$, where n is the number of vertices in the graph.*

II. Preliminaries

In this section we present known results about algebraic complexity theory that we will use in this paper. We refer to [4] for an extensive treatment of this topic.

Assume that $\mathbb{F}$ is an arbitrary field. Let $U = \mathbb{F}^u$, $V = \mathbb{F}^v$ and $W = \mathbb{F}^w$ be three vector spaces over $\mathbb{F}$, where u, v and w are three positive integers. A tensor t of format (u, v, w), also called a trilinear form of format (u, v, w), is an element of $U \otimes V \otimes W = \mathbb{F}^{u \times v \times w}$, where $\otimes$ denotes the tensor product. If we fix bases $\{x_i\}$, $\{y_j\}$ and $\{z_k\}$ of U, V and W, respectively, then we can express t as

$$t = \sum_{ijk} t_{ijk}\, x_i \otimes y_j \otimes z_k$$

for coefficients t_{ijk} in $\mathbb{F}$. The tensor t can then be represented by the 3-dimensional array $[t_{ijk}]$. We will often write $x_i \otimes y_j \otimes z_j$ simply as $x_i y_j z_k$.

The tensor corresponding to the matrix multiplication of an $m \times n$ matrix by an $n \times p$ matrix is the tensor of format $(m \times n, n \times p, m \times p)$ with coefficients $t_{ijk} = 1$ if $i = (r, s)$, $j = (s, t)$ and $k = (r, t)$ for some integers $(r, s, t) \in [m] \times [n] \times [p]$, and $t_{ijk} = 0$ otherwise. This tensor will be denoted by $\langle m, n, p\rangle$. Another example is the tensor $\sum_{\ell=1}^{n} x_\ell y_\ell z_\ell$ of format (n, n, n). This tensor is denoted $\langle n\rangle$ and corresponds to n independent scalar products.

An important notion is the concept of degeneration of tensors. We refer to [4] for the formal definition. Intuitively, the fact that a tensor t' is a degeneration of a tensor t, denoted $t' \trianglelefteq t$, means that an algorithm computing t can be converted into an "approximate algorithm" computing t' with essentially the same complexity. The notion of degeneration can be used to define the notion of border rank: the border rank of a tensor t, denoted by $\underline{R}(t)$, is the minimal $r \in \mathbb{N}$ such that $t \trianglelefteq \langle r\rangle$.

Let $t \in U \otimes V \otimes W$ and $t' \in U' \otimes V' \otimes W'$ be two tensors. We can naturally define the direct sum $t \oplus t'$, which is a tensor in $(U \oplus U') \otimes (V \oplus V') \otimes (W \oplus W')$, and the tensor product $t \otimes t'$, which is a tensor in $(U \otimes U') \otimes (V \otimes V') \otimes (W \otimes W')$. For any integer $c \geq 1$, we will denote the tensor $t \oplus \cdots \oplus t$ (with c occurrences of t) by $c \cdot t$ and the tensor $t \otimes \cdots \otimes t$ (with c occurrences of t) by $t^{\otimes c}$.

Schönhage's asymptotic sum inequality [20] will be one of the main tools used to prove our bounds. Its original statement is for estimating the exponent of square matrix multiplication, but it can be easily generalized to estimate the exponent of rectangular matrix multiplication as well. We will use the following form, which has been also used implicitly in [12], [15]. A proof can be found in [16].

Theorem 3 (Schönhage's asymptotic sum inequality)**.** *Let k, m and c be three positive integers. Let t be a tensor such that $c \cdot \langle m, m, m^k\rangle \trianglelefteq t$. Then $c \cdot m^{\omega(1,1,k)} \leq \underline{R}(t)$.*

Theorem 3 states that, if the form t can be degenerated into a direct sum of c forms, each being isomorphic to $\langle m, m, m^k\rangle$, then the inequality $c \cdot m^{w(1,1,k)} \leq \underline{R}(t)$ holds.

Let $t \in U \otimes V \otimes W$ be a tensor. Suppose that U, V and W decompose as direct sums of subspaces as follows:

$$U = \bigoplus_{i \in S_U} U_i, \quad V = \bigoplus_{j \in S_V} V_j, \quad W = \bigoplus_{k \in S_W} W_k.$$

Denote by D this decomposition. We say that t is a $\mathscr{C}$-tensor with respect to D if t can be written as

$$t = \sum_{(i,j,k) \in S_U \times S_V \times S_W} t_{ijk}$$

where each t_{ijk} is a tensor in $U_i \otimes V_j \otimes W_k$. The support of t is defined as

$$\mathrm{supp}_D(t) = \{(i, j, k) \in S_U \times S_V \times S_W \mid t_{ijk} \neq 0\},$$

and the nonzero t_{ijk}'s are called the components of t. We will usually omit the reference to D when there is no ambiguity or when the decomposition does not matter.

As a simple example, consider the complete decompositions of the spaces $U = \mathbb{F}^{m \times n}$, $V = \mathbb{F}^{n \times p}$ and $W = \mathbb{F}^{m \times p}$ (i.e., their decomposition as direct sums of one-dimensional subspaces, each subspace being spanned by one element of their basis). With respect to this decomposition, the tensor of matrix multiplication $\langle m, n, p\rangle$ is a $\mathscr{C}$-tensor with support

$$\mathrm{supp}_{\mathrm{c}}(\langle m, n, p\rangle) = \{((r, s), (s, t), (r, t)) \mid (r, s, t) \in [m] \times [n] \times [p]\}$$

where each component is trivial (i.e., isomorphic to $\langle 1, 1, 1\rangle$). In this paper the notation $\mathrm{supp}_{\mathrm{c}}(\langle m, n, p\rangle)$ will always refer to the support of $\langle m, n, p\rangle$ with respect to this complete decomposition.

We now introduce the concept of combinatorial degeneration. A subset Δ of $S_U \times S_V \times S_W$ is called diagonal if the three projections $\Delta \to S_U$, $\Delta \to S_V$ and $\Delta \to S_W$ are injective. Let Φ be a subset of $S_U \times S_V \times S_W$. A set $\Psi \subseteq \Phi$ is a combinatorial degeneration of Φ if there exists three functions $a\colon S_U \to \mathbb{Z}$, $b\colon S_V \to \mathbb{Z}$ and $c\colon S_W \to \mathbb{Z}$ such that

- for all $(i, j, k) \in \Psi$, $a(i) + b(j) + c(k) = 0$;
- for all $(i, j, k) \in \Phi \backslash \Psi$, $a(i) + b(j) + c(k) > 0$.

The most useful application of combinatorial degeneration will be the following result, which essentially states that a sum, over indices in a diagonal combinatorial degeneration of $\mathrm{supp}_D(t)$, of the components t_{ijk} is direct.

Proposition 1 (Proposition 15.30 in [4])**.** *Let t be $\mathscr{C}$-tensor with support $\mathrm{supp}_D(t)$ and components t_{ijk}. Let $\Delta \subseteq \mathrm{supp}_D(t)$ be a combinatorial degeneration of $\mathrm{supp}_D(t)$ and assume that Δ is diagonal. Then $\bigoplus_{(i,j,k) \in \Delta} t_{ijk} \trianglelefteq t$.*

When the support of t is isomorphic to $\mathrm{supp}_{\mathrm{c}}(\langle e, h, \ell\rangle)$ for some positive integers e, h and ℓ, a powerful tool to construct large diagonal combinatorial degenerations is given by the following result by Strassen (Theorem 6.6 in [22]), restated in our terminology.

Proposition 2 ([22]). *Let e_1, e_2 and e_3 be three positive integers such that $e_1 \leq e_2 \leq e_3$. For any permutation σ of $\{1,2,3\}$, there exists a diagonal set $\Delta \subseteq \mathrm{supp}_{\mathrm{c}}(\langle e_{\sigma(1)}, e_{\sigma(2)}, e_{\sigma(3)}\rangle)$ with $|\Delta| \geq \lceil 3e_1e_2/4 \rceil$ that is a combinatorial degeneration of $\mathrm{supp}_{\mathrm{c}}(\langle e_{\sigma(1)}, e_{\sigma(2)}, e_{\sigma(3)}\rangle)$.*

III. Coppersmith-Winograd's construction

In this section we describe the construction by Coppersmith and Winograd [9], which we will use as the basis of our algorithm, and several of its properties.

Section 7 of [9] describes a trilinear form F_q, where $q \in \mathbb{N}$ is a parameter, that is used to obtain the upper bound $\omega < 2.38719$. Section 8 of [9] shows how the tensor product of F_q by itself, which has border rank $\underline{R}(F_q \otimes F_q) \leq (q+2)^2$, can be used to obtain a sum of fifteen trilinear forms:

$$\sum_{\substack{0 \leq i,j,k \leq 4 \\ i+j+k=4}} T_{ijk} \trianglelefteq F_q \otimes F_q,$$

where

$$\begin{aligned}
T_{004} &= x^0_{0,0} y^0_{0,0} z^4_{q+1,q+1} \\
T_{013} &= \sum_{i=1}^{q} x^0_{0,0} y^1_{i,0} z^3_{i,q+1} + \sum_{k=1}^{q} x^0_{0,0} y^1_{0,k} z^3_{q+1,k} \\
T_{022} &= x^0_{0,0} y^2_{q+1,0} z^2_{0,q+1} + x^0_{0,0} y^2_{0,q+1} z^2_{q+1,0} + \\
&\quad \sum_{i,k=1}^{q} x^0_{0,0} y^2_{i,k} z^2_{i,k} \\
T_{112} &= \sum_{i=1}^{q} x^1_{i,0} y^1_{i,0} z^2_{0,q+1} + \sum_{k=1}^{q} x^1_{0,k} y^1_{0,k} z^2_{q+1,0} + \\
&\quad \sum_{i,k=1}^{q} x^1_{i,0} y^1_{0,k} z^2_{i,k} + \sum_{i,k=1}^{q} x^1_{0,k} y^1_{i,0} z^2_{i,k}
\end{aligned}$$

and the other eleven terms are obtained by permuting the indexes of the x-variables, the y-variables and z-variables in the above expressions (e.g., $T_{040} = x^0_{0,0} y^4_{q+1,q+1} z^0_{0,0}$ and $T_{400} = x^4_{q+1,q+1} y^0_{0,0} z^0_{0,0}$).

Let us describe in more details the notations used here. The number of x-variables is $(q+2)^2$. They are indexed as $x_{i,k}$, for $i, k \in \{0, 1, \ldots, q+1\}$. The superscript is assigned in the following way: the variable $x_{0,0}$ has superscript 0, the variables in $\{x_{i,0}, x_{0,k}\}_{1 \leq i,j \leq q}$ have superscript 1, the variables in $\{x_{q+1,0}, x_{i,k}, x_{0,q+1}\}_{1 \leq i,j \leq q}$ have superscript 2, the variables in $\{x_{q+1,k}, x_{i,q+1}\}_{1 \leq i,j \leq q}$ have superscript 3 and the variable $x_{q+1,q+1}$ has superscript 4. Note that the superscript is completely determined by the subscript. Similarly, the number of y-variables is $(q+2)^2$, and the number of z-variables is $(q+2)^2$ as well. The y-variables and the z-variables are assigned subscripts and superscripts exactly as for the x-variables. Observe that any term xyz that appears in T_{ijk} is such that x has superscript i, y has superscript j and z has superscript k.

We will later need to analyze all the forms T_{ijk}. It happens, as observed in [9], that most of these forms (all the forms except T_{112}, T_{121} and T_{211}) can be analyzed in a straightforward way, since they are isomorphic to the following matrix products:

$$\begin{aligned}
T_{004} \cong T_{040} \cong T_{400} &\cong \langle 1, 1, 1 \rangle \\
T_{013} \cong T_{031} &\cong \langle 1, 1, 2q \rangle \\
T_{103} \cong T_{301} &\cong \langle 2q, 1, 1 \rangle \\
T_{130} \cong T_{310} &\cong \langle 1, 2q, 1 \rangle \\
T_{022} &\cong \langle 1, 1, q^2+2 \rangle \\
T_{202} &\cong \langle q^2+2, 1, 1 \rangle \\
T_{220} &\cong \langle 1, q^2+2, 1 \rangle.
\end{aligned}$$

This can be seen from the definition of the trilinear form (or the tensor) corresponding to matrix multiplication. For example, the form T_{013} is isomorphic to the tensor $\sum_{\ell=1}^{2q} x_0 y_\ell z_\ell = \langle 1, 1, 2q \rangle$, which represents the product of a 1×1 matrix (a scalar) by a $1 \times 2q$ matrix (a row).

IV. Algorithm for Rectangular Matrix Multiplication

In this section we present our algorithm, which consists in the two algorithmic steps described in Subsections IV-B and IV-C. We first start by explaining in Subsections IV-A the construction we will use.

A. Our construction

Let $a_{004}, a_{400}, a_{013}, a_{103}, a_{301}, a_{022}, a_{202}, a_{112}, a_{211}$ be nine arbitrary positive rational numbers such that

$$\begin{aligned} 2a_{004} + a_{400} + 2a_{013} + 2a_{103} + \\ 2a_{301} + a_{022} + 2a_{202} + 2a_{112} + a_{211} = 1 \end{aligned} \tag{2}$$

and

$$a_{013} a_{202} a_{112} = a_{103} a_{022} a_{211}. \tag{3}$$

It will be convenient to define six additional numbers a_{040}, a_{031}, a_{130}, a_{310}, a_{220} and a_{121} as $a_{040} = a_{004}$, $a_{031} = a_{013}$, $a_{130} = a_{103}$, $a_{310} = a_{301}$, $a_{220} = a_{202}$ and $a_{121} = a_{112}$. Let us define rational numbers $A_0, A_1, A_2, A_3, A_4, B_0, B_1, B_2, B_3, B_4$ as follows.

$$\begin{aligned}
A_i &= \sum_{\substack{0 \leq j,k \leq 4 \\ i+j+k=4}} a_{ijk} \quad \text{for } i = 0,1,2,3,4 \\
B_j &= \sum_{\substack{0 \leq i,k \leq 4 \\ i+j+k=4}} a_{ijk} \quad \text{for } j = 0,1,2,3,4
\end{aligned}$$

Let N be a large enough positive integer such each Na_{ijk} is an integer. We raise the construction $F_q \otimes F_q$ described in Section III to the N-th power, which gives

$$\left(\sum_{\substack{0\leq i,j,k\leq 4\\ i+j+k=4}} T_{ijk}\right)^{\otimes N} \trianglelefteq (F_q\otimes F_q)^{\otimes N}.$$

The left term can be rewritten as $\sum_{IJK} T_{IJK}$, where the sum is over all triples of sequences IJK with $I,J,K\in\{0,1,2,3,4\}^N$ such that $I_\ell+J_\ell+K_\ell=4$ for all $\ell\in\{1,\ldots,N\}$. Here we use the notation $T_{IJK}=T_{I_1J_1K_1}\otimes\cdots\otimes T_{I_NJ_NK_N}$. Note that there are 15^N terms T_{IJK} in the above sum. In the tensor product the number of x-variables is $(q+2)^{2N}$. The number of y-variables and z-variables is also $(q+2)^{2N}$. Remember that in the original construction, each x-variable was indexed by a superscript in $\{0,1,2,3,4\}$. Each x-variable in the tensor product is thus indexed by a sequence of N such superscripts, i.e., by an element $I\in\{0,1,2,3,4\}^N$. The same is true for the y-variables and the z-variables. Note that the x-variables appearing in T_{IJK} have superscript I, the y-variables appearing in T_{IJK} have superscript J, and the z-variables appearing in T_{IJK} have superscript K.

The following definition will be useful in our analysis.

Definition 1. *Let $\overline{a}_{004}$, $\overline{a}_{040}$, $\overline{a}_{400}$, $\overline{a}_{013}$, $\overline{a}_{031}$, $\overline{a}_{103}$, $\overline{a}_{130}$, $\overline{a}_{301}$, $\overline{a}_{310}$, $\overline{a}_{022}$, $\overline{a}_{202}$, $\overline{a}_{220}$, $\overline{a}_{112}$, $\overline{a}_{121}$, $\overline{a}_{211}$ be fifteen nonnegative rational numbers. We say that a triple IJK is of type $[\overline{a}_{ijk}]$ if*

$$|\{\ell\in\{1,\ldots,N\}\mid I_\ell=i, J_\ell=j \text{ and } K_\ell=k\}|=\overline{a}_{ijk}N$$

for all 15 combinations of positive i,j,k with $i+j+k=4$.

With a slight abuse of notation, we will say that a form T_{IJK} is of type $[\overline{a}_{ijk}]$ if the triple IJK is of type $[\overline{a}_{ijk}]$.

B. First step

We set to zero all x-variables except those satisfying the following condition: their superscript I has exactly A_0N coordinates with value 0, A_1N coordinates with value 1, A_2N coordinates with value 2, A_3N coordinates with value 3 and A_4N coordinates with value 4. We will say that such a sequence I is of type A. There are

$$T_X=\binom{N}{A_0N,\ldots,A_4N}$$

sequences I of type A. After the zeroing operation, all forms T_{IJK} such that I is not of type A disappear (i.e., become zero).

We process the y-variables and the z-variables slightly differently. We set to zero all y-variables except those satisfying the following condition: their superscript J has exactly B_0N coordinates with value 0, B_1N coordinates with value 1, B_2N coordinates with value 2, B_3N coordinates with value 3 and B_4N coordinates with value 4. We will say that such a sequence is of type B. There are

$$T_Y=\binom{N}{B_0N,\ldots,B_4N}$$

sequences J of type B. Similarly, we set to zero all z-variables except those such that their superscript K is of type B (there are T_Y such sequences).

After these three zeroing operations, the forms T_{IJK} remaining are precisely those such that I is of type A, J is of type B, and K is of type B. Equivalently, the forms remaining are precisely the forms T_{IJK} that are of type $[\overline{a}_{ijk}]$ with fifteen numbers $\overline{a}_{ijk}$ (for all fifteen combinations of positive i,j,k such that $i+j+k=4$) satisfying the following four conditions:

$$\overline{a}_{ijk}N\in\{0,1,\ldots,N\}\quad\text{for all } i,j,k; \tag{4}$$

$$A_i=\sum_{j,k\,:\,i+j+k=4}\overline{a}_{ijk}\quad\text{for } i=0,1,2,3,4; \tag{5}$$

$$B_j=\sum_{i,k\,:\,i+j+k=4}\overline{a}_{ijk}\quad\text{for } j=0,1,2,3,4; \tag{6}$$

$$B_k=\sum_{i,j\,:\,i+j+k=4}\overline{a}_{ijk}\quad\text{for } k=0,1,2,3,4. \tag{7}$$

Let I be a fixed sequence of type A. The number of non-zero forms T_{IJK} with this sequence I as its first index is thus precisely

$$\mathcal{N}_X=\sum_{[\overline{a}_{ijk}]}\prod_{i=0}^{4}\binom{A_iN}{\{\overline{a}_{ijk}N\}_{j,k\,:\,i+j+k=4}},$$

where the sum is over all the choices of fifteen parameters $\overline{a}_{ijk}$'s satisfying conditions (4)–(7). Hereafter we are using the following notation: for any positive integer m and any set of integers $S=\{m_1,\ldots,m_s\}$ such that $m_1+\cdots+m_s=m$, we write $\binom{m}{S}=\binom{m}{m_1,m_2,\ldots,m_s}$.

For a fixed sequence J of type B, the number of non-zero forms T_{IJK} with this sequence J as its second index is

$$\mathcal{N}_Y=\sum_{[\overline{a}_{ijk}]}\prod_{j=0}^{4}\binom{B_jN}{\{\overline{a}_{ijk}N\}_{i,k\,:\,i+j+k=4}},$$

where the sum is again over all the choices of fifteen parameters $\overline{a}_{ijk}$'s satisfying conditions (5)–(7). Similarly, for a fixed sequence K of type B, the number of non-zero forms T_{IJK} with this sequence K as its third index is

$$\mathcal{N}_Z=\sum_{[\overline{a}_{ijk}]}\prod_{k=0}^{4}\binom{B_kN}{\{\overline{a}_{ijk}N\}_{i,j\,:\,i+j+k=4}}.$$

The total number of remaining triples is $T_X\mathcal{N}_X=T_Y\mathcal{N}_Y=T_Y\mathcal{N}_Z$. Note that this implies that $\mathcal{N}_Y=\mathcal{N}_Z$.

We will also be interested in the number of remaining forms T_{IJK} of type $[a_{ijk}]$. For a fixed sequence I of type

A, the number of non-zero forms T_{IJK} of type $[a_{ijk}]$ with this sequence I as its first index is

$$\mathcal{N}_X^* \;=\; \prod_{i=0}^{4} \binom{A_i N}{\{a_{ijk}N\}_{j,k\,:\,i+j+k=4}}.$$

For a fixed sequence J of type B, the number of non-zero forms T_{IJK} of type $[a_{ijk}]$ with this sequence J as its second index is

$$\mathcal{N}_Y^* \;=\; \prod_{j=0}^{4} \binom{B_j N}{\{a_{ijk}N\}_{i,k\,:\,i+j+k=4}}.$$

We have $T_X\mathcal{N}_X^* = T_Y\mathcal{N}_Y^*$.

We know that $\mathcal{N}_X^* \leq \mathcal{N}_X$ and $\mathcal{N}_Y^* \leq \mathcal{N}_Y$, by definition. It can be shown, from the condition $a_{013}a_{202}a_{112} = a_{103}a_{022}a_{211}$ we imposed on the a_{ijk}'s, that $\mathcal{N}_X$ and $\mathcal{N}_Y(=\mathcal{N}_Z)$ can actually be approximated by $\mathcal{N}_X^*$ and $\mathcal{N}_Y^*$.

Proposition 3. *$\mathcal{N}_X = O(N^8\mathcal{N}_X^*)$ and $\mathcal{N}_Y = O(N^8\mathcal{N}_Y^*)$.*

C. Second step

The first step showed how to convert the trilinear form $(F_q \otimes F_q)^{\otimes N}$ into a sum of $T_X\mathcal{N}_X$ triples. Among these triples exactly $T_X\mathcal{N}_X^*$ triples are of type $[a_{ijk}]$, which means that they are isomorphic to $\bigotimes_{i,j,k:i+j+k=4} T_{ijk}^{\otimes a_{ijk}N}$.

The sum obtained is nevertheless not direct: the triples share variables. A generalization of the pruning argument in [9] (see also [21], [23]) to our asymmetric setting shows that this sum can be converted, for any positive constant ϵ, into a *direct* sum of

$$\Omega\left(\frac{T_X\mathcal{N}_X^*}{(\mathcal{N}_X+\mathcal{N}_Y+\mathcal{N}_Z)^{1+\epsilon}}\right)$$

triples, all of type $[a_{ijk}]$, by zeroing variables.

From Stirling's approximation, we have

$$T_X = \Theta\left(\frac{1}{N^2}\left(\frac{1}{A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4}}\right)^N\right).$$

Suppose that the inequality

$$A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4} \geq B_0^{B_0}B_1^{B_1}B_2^{B_2}B_3^{B_3}B_4^{B_4} \qquad (8)$$

holds. In this case $T_X = O(T_Y)$, and then the equality $T_X\mathcal{N}_X^* = T_Y\mathcal{N}_Y^*$ implies that $\mathcal{N}_Y^* = O(\mathcal{N}_X^*)$, which, combined with Proposition 3, gives

$$\frac{T_X\mathcal{N}_X^*}{(\mathcal{N}_X+\mathcal{N}_Y+\mathcal{N}_Z)^{1+\epsilon}} = \Omega\left(\frac{T_X\mathcal{N}_X^*}{(N^8\mathcal{N}_X^*)^{1+\epsilon}}\right).$$

Finally, by using the trivial upper bound $\mathcal{N}_X^* \leq 15^N$, we obtain the following theorem.

Theorem 4. *Let q be any positive integer and a_{004}, a_{400}, a_{013}, a_{103}, a_{301}, a_{022}, a_{202}, a_{112} and a_{211} be any nine positive rational numbers satisfying Conditions (2), (3) and (8). Then, for any constant $\epsilon > 0$, the trilinear form $(F_q \otimes F_q)^{\otimes N}$ can be converted (i.e., degenerated) into a direct sum of*

$$\Omega\left(\frac{1}{N^{10+8\epsilon}15^{N\epsilon}}\left[\frac{1}{A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4}}\right]^N\right)$$

forms, each form being isomorphic to $\bigotimes_{\substack{0\leq i,j,k\leq 4\\ i+j+k=4}} T_{ijk}^{\otimes a_{ijk}N}$.

V. Upper Bounds on the Exponent of Rectangular Matrix Multiplication

Theorem 4 showed how the trilinear form $(F_q \otimes F_q)^{\otimes N}$ can be converted into a direct sum of many forms T_{IJK} such that

$$T_{IJK} \cong \bigotimes_{i,j,k:i+j+k=4} T_{ijk}^{\otimes a_{ijk}N}.$$

In order to apply Schönhage's asymptotic sum inequality (Theorem 3), we need to analyze the smaller forms T_{ijk}. In Subsection V-A we analyze the forms T_{112}, T_{121} and T_{211}. Then, in Subsection V-B, we put all our results together and prove our main result.

A. The forms T_{112}, T_{121} and T_{211}

We first focus on the form T_{211}. The following proposition states that tensor powers of T_{211} can be used to construct a direct sum of several trilinear forms, each one being a $\mathscr{C}$-tensor in which the support and all the components are isomorphic to a rectangular matrix product. Its proof, omitted here, follows the ideas of the proof of the lemma at page 270 in [9].

Proposition 4. *Let b be any constant such that $0.916027 < b \leq 1$. Then there exists a constant $c \geq 1$ depending only on b such that, for any $\epsilon > 0$ and any large enough integer m, the form $T_{211}^{\otimes 2m}$ can be converted into a direct sum of*

$$\Omega\left(\frac{1}{mc^{2\epsilon m}}\cdot\left[\frac{2}{(2b)^b(1-b)^{1-b}}\right]^{2m}\right)$$

trilinear forms, each form being a $\mathscr{C}$-tensor in which:

- *each component is isomorphic to $\langle q^{2bm}, q^{2bm}, q^{2(1-b)m}\rangle$;*
- *the support is isomorphic to $\mathrm{supp}_c(\langle 1,1,H\rangle)$, where $H = \Omega\left(\frac{1}{\sqrt{m}}\cdot\left[(2b)^b(1-b)^{(1-b)}\right]^{2m}\right)$.*

The forms T_{112} and T_{121} can be analyzed in the same way as T_{211} by permuting the roles of the x-variables, the y-variables and the z-variables. Similarly to the statement of Proposition 4, the form $T_{112}^{\otimes 2m}$ gives a direct sum of $\mathscr{C}$-tensors with support isomorphic to $\langle 1,H,1\rangle$, each component in the tensors being isomorphic to $\langle q^{2bm}, q^{2(1-b)m}, q^{2bm}\rangle$. The form $T_{121}^{\otimes 2m}$ gives a direct sum of $\mathscr{C}$-tensors with support isomorphic to $\mathrm{supp}_c(\langle H,1,1\rangle)$, each component being isomorphic to $\langle q^{2(1-b)m}, q^{2bm}, q^{2bm}\rangle$.

Suppose that different constants are used to treat each of the three forms: the forms T_{112} and T_{121} are processed with some constant b, while T_{211} is processed with another constant $\tilde{b}$. For any fixed values a_{112}, a_{211} and any $\epsilon > 0$, the form $T_{112}^{\otimes a_{112}N} \otimes T_{121}^{\otimes a_{112}N} \otimes T_{211}^{\otimes a_{211}N}$ can then be used to construct a direct sum of

$$\Omega\left(\frac{2^{(2a_{112}+a_{211})N}}{N^3 c'^{N\epsilon} \cdot [(2b)^b(1-b)^{1-b}]^{2a_{112}N} \cdot \left[(2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}}\right]^{a_{211}N}}\right)$$

$\mathscr{C}$-tensors, for some value $c' \geq 1$ depending only on b and $\tilde{b}$. Each of these $\mathscr{C}$-tensors has a support isomorphic to $\mathrm{supp_c}(\langle H_{112}, H_{112}, H_{211}\rangle)$, where

$$\begin{aligned} H_{112} &= \Omega\left(\frac{1}{\sqrt{N}} \cdot \left[(2b)^b(1-b)^{(1-b)}\right]^{a_{112}N}\right) \\ H_{211} &= \Omega\left(\frac{1}{\sqrt{N}} \cdot \left[(2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{(1-\tilde{b})}\right]^{a_{211}N}\right). \end{aligned}$$

In all these $\mathscr{C}$-tensors, each component is isomorphic to the rectangular matrix multiplication

$$\langle q^{(a_{112}+a_{211}\tilde{b})N}, q^{(a_{112}+a_{211}\tilde{b})N}, q^{(2a_{112}b+a_{211}(1-\tilde{b}))N}\rangle. \quad (9)$$

We can then use Propositions 1 and 2 to convert each $\mathscr{C}$-tensor into a direct sum of at least $\frac{3}{4}H_{112} \times \min(H_{112}, H_{211})$ trilinear forms, each isomorphic to (9). We thus obtain the following result.

Proposition 5. *Let a_{112} and a_{211} be any two positive constants. Let b and $\tilde{b}$ be any two constants such that* $0.916027 < b, \tilde{b} \leq 1$. *Define*

$$\mathcal{H} = \max\left(\left[(2b)^b(1-b)^{1-b}\right]^{a_{112}}, \left[(2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}}\right]^{a_{211}}\right).$$

Then there exists a constant $c' \geq 1$ such that, for any $\epsilon > 0$, the trilinear form $T_{112}^{\otimes a_{112}N} \otimes T_{121}^{\otimes a_{112}N} \otimes T_{211}^{\otimes a_{211}N}$ can be converted into a direct sum of

$$\Omega\left(\frac{1}{N^4 c'^{N\epsilon}} \cdot \left[\frac{2^{2a_{112}+a_{211}}}{\mathcal{H}}\right]^N\right)$$

forms, each form being isomorphic to (9).

B. Main theorem

Let us define the following three quantities.

$$\begin{aligned} Q &= (2q)^{a_{103}+a_{301}} \times (q^2+2)^{a_{202}} \times q^{a_{112}+a_{211}\tilde{b}} \\ R &= (2q)^{2a_{013}} \times (q^2+2)^{a_{022}} \times q^{2a_{112}b+(1-\tilde{b})a_{211}} \\ \mathcal{M} &= \frac{2^{2a_{112}+a_{211}}}{A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4}} \times \frac{1}{\mathcal{H}} \end{aligned}$$

Our main theorem gives an upper bound on $\omega(1,1,k)$ that depends on these quantities.

Theorem 5. *Let q be any positive integer and $b, \tilde{b}$ be such that* $0.916027 < b, \tilde{b} \leq 1$. *Let a_{004}, a_{400}, a_{013}, a_{103}, a_{301}, a_{022}, a_{202}, a_{112} and a_{211} be any nine positive rational numbers satisfying Conditions (2), (3) and (8). Then*

$$\mathcal{M}Q^{w(1,1,\frac{\log R}{\log Q})} \leq (q+2)^2.$$

Proof: Let $\epsilon > 0$ be an arbitrary positive value. Let N be a large integer and consider the trilinear form $(F_q \otimes F_q)^{\otimes N}$. Theorem 4 shows that this form can be used to obtain a direct sum of

$$r_1 = \Omega\left(\frac{1}{N^{10+8\epsilon}15^{N\epsilon}}\left[\frac{1}{A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4}}\right]^N\right)$$

forms, each isomorphic to $\bigotimes_{i,j,k\colon i+j+k=4} T_{ijk}^{\otimes a_{ijk}N}$.

All the terms T_{ijk} in this form, except T_{112}, T_{121} and T_{211}, correspond to matrix multiplications and have been analyzed in Section III. By Proposition 5 the part $T_{112}^{\otimes a_{112}N} \otimes T_{121}^{\otimes a_{112}N} \otimes T_{211}^{\otimes a_{211}N}$ can be used to obtain a direct sum of

$$r_2 = \Omega\left(\frac{1}{N^4 c'^{N\epsilon}} \cdot \left[\frac{2^{2a_{112}+a_{211}}}{\mathcal{H}}\right]^N\right)$$

matrix multiplications

$$\langle q^{(a_{112}+a_{211}\tilde{b})N}, q^{(a_{112}+a_{211}\tilde{b})N}, q^{(2a_{112}b+(1-\tilde{b})a_{211})N}\rangle.$$

This means that the trilinear form $(F_q \otimes F_q)^{\otimes N}$ can be converted into a direct sum of $r_1 r_2$ matrix multiplications $\langle Q^N, Q^N, R^N\rangle$. In other words:

$$r_1 r_2 \cdot \langle Q^N, Q^N, R^N\rangle \trianglelefteq (F_q \otimes F_q)^{\otimes N}.$$

Since $\underline{R}(F_q \otimes F_q) \leq (q+2)^2$, as mentioned in Section III, we know that $\underline{R}\left((F_q \otimes F_q)^{\otimes N}\right) \leq (q+2)^{2N}$. By Schönhage's asymptotic sum inequality (Theorem 3) we then conclude that

$$r_1 r_2 \times Q^{N\omega(1,1,\frac{\log R}{\log Q})} \leq (q+2)^{2N}.$$

Taking the N-th root, we obtain:

$$\frac{1}{(15c')^{\epsilon}N^{(14+8\epsilon)/N}} \times \mathcal{M}Q^{\omega(1,1,\frac{\log R}{\log Q})} \quad \leq \quad (q+2)^2.$$

By letting N grow to infinity, and then letting ϵ decrease to zero, we conclude that $\mathcal{M}Q^{\omega(1,1,\frac{\log R}{\log Q})} \leq (q+2)^2$. ■

VI. Optimization

In this section we use Theorem 5 to derive numerical upper bounds on the exponent of rectangular matrix multiplication, and prove Theorem 1.

q	5	6
b	0.984599222	0.94866036
$\tilde{b}$	0.919886704	0.99996514
a_{400}	0.004942000	0.00000090
a_{103}	0.010965995	0.01553556
a_{301}	0.055710210	0.00079349
a_{022}	0.037622078	0.22704392
a_{202}	0.138698196	0.05836108
a_{112}	0.145715589	0.20388121
a_{211}	0.245013049	0.13394891
$\log R/\log Q$	0.530200005...	2.00000004...
$(2\log(q+2)-\log\mathcal{M})/\log Q$	2.060395...	3.256688...

Table II
TWO SOLUTIONS FOR OUR OPTIMIZATION PROBLEM. THE FIRST TEN ROWS GIVE (EXACT) VALUES OF THE TEN PARAMETERS. THE NUMERICAL VALUES OF THE LAST TWO ROWS SHOW THAT $\omega(1,1,0.5302) < 2.060396$, AND $\omega(1,1,2) < 3.256689$.

A. Rectangular matrix multiplication

We first explain how to use Theorem 5 to derive an upper bound on $\omega(1,1,k)$ for an arbitrary value k, and show how to obtain the results stated in Table I and Figure 1.

We use the following strategy. We take a positive integer q, seven positive rational numbers a_{400}, a_{103}, a_{301}, a_{022}, a_{202}, a_{112} and a_{211}, and two values $b, \tilde{b}$ such that $0.916027 < b, \tilde{b} \leq 1$. We then fix

$$a_{013} = \frac{a_{103}a_{022}a_{211}}{a_{202}a_{112}}$$

and $a_{004} = \frac{1-(a_{400}+a_{022}+a_{211})-2(a_{013}+a_{103}+a_{301}+a_{202}+a_{112})}{2}$. The conditions that have to be satisfied are $0 < a_{004}, a_{013} \leq 1$ and

$$A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4} \geq B_0^{B_0}B_1^{B_1}B_2^{B_2}B_3^{B_3}B_4^{B_4}.$$

If these conditions are satisfied, by Theorem 5 this gives the upper bound

$$\omega\left(1,1,\frac{\log R}{\log Q}\right) \leq \frac{2\log(q+2)-\log\mathcal{M}}{\log Q}.$$

The problem of finding an upper bound on $\omega(1,1,k)$ is thus reduced to solving a nonlinear optimization problem. The upper bounds presented in Table I are obtained precisely by solving this optimization problem. For instance, we show exact values of the parameters proving that $\omega(1,1,0.5302) < 2.060396$ and $\omega(1,1,2) < 3.256689$ in Table II.

B. The value α

We now describe how to use Theorem 5 to obtain a lower bound on the value α. The analysis is more delicate than in the previous subsection, since we need to exhibit parameters such that $\mathcal{M}Q^2 = (q+2)^2$, with an equality rather than an inequality, and is done by finding analytically the optimal values of all but a few parameters.

Let q be an integer such that $q \geq 5$. For convenience, we will write $\kappa = 1/(q+2)^2$. Let a_{112} and a_{211} be any rational numbers such that $0 < a_{112} < q\kappa$ and $0 < a_{211} < (q^2+2)\kappa$. We set the parameters b, $\tilde{b}$, a_{004}, a_{103}, a_{202} and a_{301} as follows: $b = 1$, $\tilde{b} = q^2/(q^2+2)$, $a_{400} = \kappa$, $a_{103} = q\kappa - a_{112}$, $a_{202} = \left((q^2+2)\kappa - a_{211}\right)/2$ and $a_{301} = q\kappa$.

Putting these values in the formula for Q, we obtain:

$$\begin{aligned} Q &= (2q)^{q\kappa+a_{103}} \times \left(q^2+2\right)^{\frac{(q^2+2)\kappa-a_{211}}{2}} \times q^{a_{112}+q^2a_{211}/(q^2+2)} \\ &= (2q)^{2q\kappa} \times (q^2+2)^{\frac{(q^2+2)\kappa}{2}} \times 2^{-a_{112}} \times \left(\frac{q^{q^2/(q^2+2)}}{\sqrt{q^2+2}}\right)^{a_{211}}. \end{aligned}$$

Observe that $A_1 = A_3 = 2q\kappa$, $A_2 = (q^2+2)\kappa$, $A_4 = \kappa$ and $A_0 = 1-(A_1+A_2+A_3+A_4) = \kappa$. Then we obtain the following equality.

$$\frac{1}{A_0^{A_0}A_1^{A_1}A_2^{A_2}A_3^{A_3}A_4^{A_4}} = \frac{(q+2)^2}{(2q)^{4q\kappa}(q^2+2)^{(q^2+2)\kappa}}$$

The following proposition shows that, when a_{112} is small enough, the condition $\mathcal{M}Q^2 = (q+2)^2$ is satisfied.

Proposition 6. *Suppose that*

$$a_{112} \leq \left(1 + \frac{2q^2}{q^2+2}\log_2(q) - \log_2(q^2+2)\right)a_{211}. \quad (10)$$

Then $\mathcal{M}Q^2 = (q+2)^2$.

Proof: Our choice for b and $\tilde{b}$ gives

$$\begin{aligned} \left[(2b)^b(1-b)^{1-b}\right]^{a_{112}} &= 2^{a_{112}} \\ \left[(2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}}\right]^{a_{211}} &= \left[\frac{2}{q^2+2}\cdot q^{\frac{2q^2}{q^2+2}}\right]^{a_{211}}. \end{aligned}$$

Inequality (10) then implies that $\left[(2b)^b(1-b)^{1-b}\right]^{a_{112}} \leq \left[(2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}}\right]^{a_{211}}$. In consequence,

$$\mathcal{M} = \frac{(q+2)^2}{(2q)^{4q\kappa}(q^2+2)^{(q^2+2)\kappa}} \times 4^{a_{112}} \times \left[\frac{q^2+2}{q^{2q^2/(q^2+2)}}\right]^{a_{211}},$$

which gives $\mathcal{M}Q^2 = (q+2)^2$. ■

We now explain how to determine the three remaining parameters a_{004}, a_{013} and a_{022}. Remember that the parameters should satisfy the equalities

$$a_{013} = \frac{a_{103}a_{211}}{a_{202}a_{112}}a_{022}$$

and $2a_{004}+a_{400}+2a_{013}+2a_{103}+2a_{301}+a_{022}+2a_{202}+2a_{112}+a_{211} = 1$. From our choice of parameters, the second equality can be rewritten as $2a_{004}+2a_{013}+a_{022} = \kappa$. Since the parameter a_{004} should be positive, we obtain the condition

$$\left(\frac{4(q\kappa-a_{112})a_{211}}{((q^2+2)\kappa-a_{211})a_{112}}+1\right)a_{022} < \kappa. \quad (11)$$

If a_{022}, a_{112} and a_{211} satisfy this inequality, then the parameter a_{004} is fixed:

$$a_{004} = \left(\kappa - \left(\frac{4(q\kappa-a_{112})a_{211}}{((q^2+2)\kappa-a_{211})a_{112}}+1\right)a_{022}\right)/2.$$

All the values are thus determined by the choice of q, a_{022}, a_{112} and a_{211}. We then want to solve the following optimization problem.

Maximize $\frac{\log R}{\log Q}$ subject to

- $0 \leq a_{022} \leq 1$;
- $0 < a_{112} \leq 5\kappa$;
- $0 \leq a_{211} \leq (q^2+2)\kappa$;
- q is an integer such that $q \geq 5$;
- Inequalities (10) and (11) hold;
- $\frac{(2q)^{4q\kappa}(q^2+2)^{(q^2+2)\kappa}}{(q+2)^2} \geq B_0^{B_0} B_1^{B_1} B_2^{B_2} B_3^{B_3} B_4^{B_4}$.

By taking $q = 5$, $a_{022} = 0.0174853$, $a_{112} = 0.0945442$ and $a_{211} = 0.1773724$, we obtain the value

$$\alpha \geq \frac{\log R}{\log Q} > 0.30298.$$

Acknowledgment

The author is grateful to Virginia Vassilevska Williams and Ryan Williams for helpful correspondence about Andrew Stothers' work, and to Virginia Vassilevska Williams for suggesting that the next step is to use higher tensor powers of the basic construction to improve rectangular matrix multiplication. He also acknowledges support from the JSPS and the MEXT, under the grant-in-aids Nos. 22800006 and 24700005.

References

[1] N. Alon, A. Shpilka, and C. Umans, "On sunflowers and matrix multiplication," in *Proceedings of the 27th Conference on Computational Complexity*, 2012, pp. 214 – 223.

[2] N. Alon and R. Yuster, "Fast algorithms for maximum subset matching and all-pairs shortest paths in graphs with a (not so) small vertex cover," in *Proceedings of the 15th Annual European Symposium on Algorithms*, 2007, pp. 175–186.

[3] R. R. Amossen and R. Pagh, "Faster join-projects and sparse matrix multiplications," in *Proceedings of the 12th International Conference on Database Theory*, 2009, pp. 121–126.

[4] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*. Springer, 1997.

[5] H. Cohn, R. D. Kleinberg, B. Szegedy, and C. Umans, "Group-theoretic algorithms for matrix multiplication," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005, pp. 379–388.

[6] H. Cohn and C. Umans, "A group-theoretic approach to fast matrix multiplication," in *Proceedings of the 44th Symposium on Foundations of Computer Science*, 2003, pp. 438–449.

[7] D. Coppersmith, "Rapid multiplication of rectangular matrices," *SIAM Journal on Computing*, vol. 11, no. 3, pp. 467–471, 1982.

[8] ——, "Rectangular matrix multiplication revisited," *Journal of Complexity*, vol. 13, no. 1, pp. 42–49, 1997.

[9] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of Symbolic Computation*, vol. 9, no. 3, pp. 251–280, 1990.

[10] A. Czumaj, M. Kowaluk, and A. Lingas, "Faster algorithms for finding lowest common ancestors in directed acyclic graphs," *Theoretical Computer Science*, vol. 380, no. 1-2, pp. 37–46, 2007.

[11] C. Demetrescu and G. F. Italiano, "Fully dynamic transitive closure: Breaking through the $o(n^2)$ barrier," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000, pp. 381–389.

[12] X. Huang and V. Y. Pan, "Fast rectangular matrix multiplication and applications," *Journal of Complexity*, vol. 14, no. 2, pp. 257–299, 1998.

[13] H. Kaplan, N. Rubin, M. Sharir, and E. Verbin, "Counting colors in boxes," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 785–794.

[14] H. Kaplan, M. Sharir, and E. Verbin, "Colored intersection searching via sparse rectangular matrix multiplication," in *Proceedings of the 22nd ACM Symposium on Computational Geometry*, 2006, pp. 52–60.

[15] S. Ke, B. Zeng, W. Han, and V. Y. Pan, "Fast rectangular matrix multiplication and some applications," *Science in China Series A: Mathematics*, vol. 51, no. 3, pp. 389–406, 2008.

[16] G. Lotti and F. Romani, "On the asymptotic complexity of rectangular matrix multiplication," *Theoretical Computer Science*, vol. 23, pp. 171–185, 1983.

[17] M. Patrascu and R. Williams, "On the possibility of faster SAT algorithms," in *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010, pp. 1065–1075.

[18] L. Roditty and A. Shapira, "All-pairs shortest paths with a sublinear additive error," *ACM Transactions on Algorithms*, vol. 7, no. 4, p. 45, 2011.

[19] P. Sankowski and M. Mucha, "Fast dynamic transitive closure with lookahead," *Algorithmica*, vol. 56, no. 2, pp. 180–197, 2010.

[20] A. Schönhage, "Partial and total matrix multiplication," *SIAM Journal on Computing*, vol. 10, no. 3, pp. 434–455, 1981.

[21] A. Stothers, "On the complexity of matrix multiplication," Ph.D. dissertation, University of Edinburgh, 2010.

[22] V. Strassen, "Relative bilinear complexity and matrix multiplication," *Journal für die reine und angewandte Mathematik*, vol. 375-376, pp. 406–443, 1987.

[23] V. Vassilevska Williams, "Multiplying matrices faster than Coppersmith-Winograd," in *Proceedings of the 44th ACM Symposium on Theory of Computing*, 2012, pp. 887–898.

[24] R. Williams, "Non-uniform ACC circuit lower bounds," in *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011, pp. 115–125.

[25] R. Yuster, "Efficient algorithms on sets of permutations, dominance, and real-weighted APSP," in *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009, pp. 950–957.

[26] R. Yuster and U. Zwick, "Detecting short directed cycles using rectangular matrix multiplication and dynamic programming," in *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004, pp. 254–260.

[27] ——, "Fast sparse matrix multiplication," *ACM Transactions on Algorithms*, vol. 1, no. 1, pp. 2–13, 2005.

[28] U. Zwick, "All pairs lightest shortest paths," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999, pp. 61–69.

[29] ——, "All pairs shortest paths using bridging sets and rectangular matrix multiplication," *Journal of the ACM*, vol. 49, no. 3, pp. 289–317, 2002.