Powers of Tensors and Fast Matrix Multiplication

François Le Gall
Department of Computer Science
The University of Tokyo
Hongo 7-3-1, Bunkyu-ku, Tokyo, Japan
legall@is.s.u-tokyo.ac.jp

ABSTRACT

This paper presents a method to analyze the powers of a given trilinear form (a special kind of algebraic construction also called a tensor) and obtain upper bounds on the asymptotic complexity of matrix multiplication. Compared with existing approaches, this method is based on convex optimization, and thus has polynomial-time complexity. As an application, we use this method to study powers of the construction given by Coppersmith and Winograd [Journal of Symbolic Computation, 1990] and obtain the upper bound $\omega < 2.3728639$ on the exponent of square matrix multiplication, which slightly improves the best known upper bound.

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*computations on matrices*; I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*algebraic algorithms, analysis of algorithms*

General Terms

Algorithms, Theory

Keywords

Algebraic complexity theory, matrix multiplication

1. INTRODUCTION

Matrix multiplication is one of the most fundamental tasks in mathematics and computer science. While the product of two $n \times n$ matrices over a field can naturally be computed in $O(n^3)$ arithmetic operations, Strassen showed in 1969 that $O(n^{2.81})$ arithmetic operations are enough [15]. The discovery of this algorithm for matrix multiplication with subcubic complexity gave rise to a new area of research, where the central question is to determine the value of the exponent of square matrix multiplication, denoted ω, and defined as the minimal value such that two $n \times n$ matrices over a field can be multiplied using $O(n^{\omega+\varepsilon})$ arithmetic operations for any $\varepsilon > 0$. It has been widely conjectured that $\omega = 2$ and several conjectures in combinatorics and group theory, if true, would lead to this result [1, 6, 7, 8]. However, the best upper bound obtained so far is $\omega < 2.38$, as we explain below.

Coppersmith and Winograd [8] showed in 1987 that $\omega < 2.3754770$. Their approach can be described as follows. A trilinear form is, informally speaking, a three-dimensional array with coefficients in a field $\mathbb{F}$. For any trilinear form t one can define its border rank, denoted $\underline{R}(t)$, which is a positive integer characterizing the number of arithmetic operations needed to compute the form. For any trilinear form t and any real number $\rho \in [2, 3]$, one can define a real number $V_\rho(t)$, called the *value* of the trilinear form. The theory developed by Schönhage [13] shows that, for any $m \geq 1$ and any $\rho \in [2, 3]$, the following statement hold:

$$\left(V_\rho(t^{\otimes m})\right)^{1/m} \geq \underline{R}(t) \implies \omega \leq \rho. \qquad (1)$$

Here the notation $t^{\otimes m}$ represents the trilinear form obtained by taking the m-th tensor power of t. Coppersmith and Winograd presented a specific trilinear form $\mathfrak{t}$, obtained by modifying a construction given earlier by Strassen [16], computed its border rank $\underline{R}(\mathfrak{t})$, and introduced deep techniques to estimate the value $V_\rho(\mathfrak{t})$. In particular, they showed how a lower bound $\tilde{V}_\rho(\mathfrak{t})$ on $V_\rho(\mathfrak{t})$ can be obtained for any $\rho \in [2, 3]$ by solving an optimization problem. Solving this optimization problem, they obtained the upper bound $\omega < 2.3871900$, via Statement (1) with $t = \mathfrak{t}$ and $m = 1$, by finding the smallest ρ such that $\tilde{V}_\rho(\mathfrak{t}) \geq \underline{R}(\mathfrak{t})$. They then proceeded to study the tensor power $\mathfrak{t}^{\otimes 2}$ and showed that, despite several new technical difficulties, a similar approach can be used to reduce the computation of a lower bound $\tilde{V}_\rho(\mathfrak{t}^{\otimes 2})$ on $V_\rho(\mathfrak{t}^{\otimes 2})$ to solving another optimization problem of several variables. They discovered that $\tilde{V}_\rho(\mathfrak{t}^{\otimes 2}) > [\tilde{V}_\rho(\mathfrak{t})]^2$, due to the fact that the analysis of $\mathfrak{t}^{\otimes 2}$ was finer, thus giving a better upper bound on ω via Statement (1) with $t = \mathfrak{t}$ and $m = 2$. Solving numerically the new optimization problem, they obtained the upper bound $\omega < 2.3754770$.

In view of the improvement obtained by taking the second tensor power, a natural question was to investigate higher powers of the construction $\mathfrak{t}$ by Coppersmith and Winograph. Investigating the third power was explicitly mentioned as an open problem in [8]. More that twenty years later, Stothers showed that, while the third power does not seem to lead to any improvement, the fourth power does give an improvement [14] (see also [9]). The improvement was obtained again via Statement (1), by showing how to

ISSAC '14, July 23 - 25, 2014, Kobe, Japan
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2501-1/14/07 ...$15.00.
http://dx.doi.org/10.1145/2608628.2608664.

reduce the computation of $V_\rho(\mathfrak{t}^{\otimes 4})$ to solving a non-convex optimization problem. The upper bound $\omega < 2.3736898$ was obtained in [9, 14] by finding numerically a solution of this optimization problem. It was later discovered that this solution was not optimal, and the improved upper bound $\omega < 2.3729269$ was given in [17] by exhibiting a better solution of the same optimization problem. Independently, Vassilevska Williams [17] constructed a powerful and general framework to analyze recursively powers of a class of trilinear forms, including the trilinear form $\mathfrak{t}$ by Coppersmith and Winograd, and showed how to automatically reduce, for any form t in this class and any integer $m \geq 2$, the problem of obtaining lower bounds on $V_\rho(t^{\otimes m})$ to solving (in general non-convex) optimization problems. The upper bound $\omega < 2.3729$ was obtained [17] by applying this framework with $t = \mathfrak{t}$ and $m = 8$, and numerically solving this optimization problem.[1] A natural question is to determine what bounds on ω can be obtained by studying $\mathfrak{t}^{\otimes m}$ for $m > 8$. One may even hope that, when m goes to infinity, the upper bound on ω goes to two. Unfortunately, this question can hardly be answered by this approach since the optimization problems are highly non-convex and become intractable even for modest values of m.

In this paper we show how to modify the framework developed in [17] in such a way that the computation of $V_\rho(\mathfrak{t}^{\otimes m})$ reduces to solving poly(m) instances of *convex* optimization problems, each having poly(m) variables. From a theoretical point a view, since a solution of such convex problems can be found in polynomial time, via Statement (1) we obtain an algorithm to derive an upper bound on ω from $\mathfrak{t}^{\otimes m}$ in time polynomial in m. From a practical point of view, the convex problems we obtain can also be solved efficiently, and have several desirable properties (in particular, the optimality of a solution can be guaranteed by using the dual problem). We use this method to analyze $\mathfrak{t}^{\otimes 16}$ and $\mathfrak{t}^{\otimes 32}$, and obtain the new upper bounds on ω described in Table 1. Besides leading to an improvement for ω, these results strongly suggest that studying powers higher than 32 will give only negligible improvements.

Our method is actually more general and can be used to efficiently obtain lower bounds on $V_\rho(t \otimes t')$ for any trilinear forms t and t' that have a structure "similar" to $\mathfrak{t}$. This applies in particular to the case where t' is trivial, giving an efficient (i.e., based on convex optimization) way to compute lower bounds on $V_\rho(t)$. Indeed, considering possible future applications of our approach, we have been attentive of stating our techniques as generally as possible.

2. ALGEBRAIC COMPLEXITY THEORY

This section presents the notions of algebraic complexity needed for this work. We refer to, e.g., [3, 5] for more detailed treatments. In this paper $\mathbb{F}$ denotes an arbitrary field.

2.1 Trilinear forms

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

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

for coefficients t_{ijk} in $\mathbb{F}$. We will usually write $x_i \otimes y_j \otimes z_j$ simply as $x_i y_j z_k$.

Table 1: Upper bounds on ω obtained by analyzing the m-th power of the construction $\mathfrak{t}$ by Coppersmith and Winograd.

m	Upper bound	Reference
1	$\omega < 2.3871900$	Ref. [8]
2	$\omega < 2.3754770$	Ref. [8]
4	$\omega < 2.3729269$	Ref. [17]
8	$\omega < 2.3728642$	this paper (Section 6.3) ($\omega < 2.3729$ given in Ref. [17])
16	$\omega < 2.3728640$	this paper (Section 6.3)
32	$\omega < 2.3728639$	this paper (Section 6.3)

Matrix multiplication of an $m \times n$ matrix with entries in $\mathbb{F}$ by an $n \times p$ matrix with entries in $\mathbb{F}$ corresponds to the trilinear form on $(\mathbb{F}^{m\times n}, \mathbb{F}^{n\times p}, \mathbb{F}^{m\times p})$ with coefficients $t_{ijk} = 1$ if $i = (r,s)$, $j = (s,t)$ and $k = (r,t)$ for some integers $(r,s,t) \in \{1,\ldots,m\} \times \{1,\ldots,n\} \times \{1,\ldots,p\}$, and $t_{ijk} = 0$ otherwise. Indeed, this form can be rewritten as

$$\sum_{r=1}^{m}\sum_{t=1}^{p}\left(\sum_{s=1}^{n} x_{(r,s)} y_{(s,t)}\right) z_{(r,t)}.$$

Then, replacing the x-variables by the entries of the first matrix and the y-variables by the entries of the second matrix, the coefficient of $z_{(r,t)}$ in the above expression represents the entry in the r-th row and the t-th column of the matrix product of these two matrices. This trilinear form will be denoted by $\langle m, n, p\rangle$.

Another important example is the form $\sum_{\ell=1}^{n} x_\ell y_\ell z_\ell$. This trilinear form on $(\mathbb{F}^n, \mathbb{F}^n, \mathbb{F}^n)$ is denoted $\langle n\rangle$ and corresponds to n independent scalar products.

Given a tensor $t \in U \otimes V \otimes W$, it will be convenient to denote by t_{C} and t_{C^2} the tensors in $V \otimes W \otimes U$ and $W \otimes U \otimes V$, respectively, obtained by permuting cyclicly the coordinates of t: $t_{\mathsf{C}} = \sum_{ijk} t_{ijk} y_j \otimes z_k \otimes x_i$ and $t_{\mathsf{C}^2} = \sum_{ijk} t_{ijk} z_k \otimes x_i \otimes y_j$.

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

Let λ be an indeterminate and consider the extension $\mathbb{F}[\lambda]$ of $\mathbb{F}$, i.e., the set of all polynomials over $\mathbb{F}$ in λ. Let $t \in \mathbb{F}^{u\times v\times w}$ and $t' \in \mathbb{F}^{u'\times v'\times w'}$ be two tensors. We say that t' is a degeneration of t, denoted $t' \trianglelefteq t$, if there exist three matrices $\alpha \in \mathbb{F}[\lambda]^{u'\times u}$, $\beta \in \mathbb{F}[\lambda]^{v'\times v}$, $\gamma \in \mathbb{F}[\lambda]^{w'\times w}$ such that

$$\lambda^s t' + \lambda^{s+1} t'' = \sum_{ijk} t_{ijk}\, \alpha(x_i) \otimes \beta(y_j) \otimes \gamma(z_j)$$

[1] Note that, while the upper bound on ω obtained for the eighth power is stated as $\omega < 2.3727$ in the conference version [17], the statement has been corrected to $\omega < 2.3729$ in the most recent version (available at the author's homepage), since the previous bound omitted some necessary constraints in the optimization problem. Our results confirm the value of the latter bound, and increase its precision.

for some tensor $t'' \in \mathbb{F}[\lambda]^{u' \times v' \times w'}$ and some nonnegative integer s. Intuitively, the fact that a tensor t' is a degeneration of a tensor t means that an algorithm computing t can be converted into another algorithm computing t' with essentially the same complexity. The notion of degeneration can be used to define the notion of border rank of a tensor t, denoted $\underline{R}(t)$, as follows:

$$\underline{R}(t) = \min\{r \in \mathbb{N} \mid t \trianglelefteq \langle r \rangle\}.$$

The border rank is submultiplicative: $\underline{R}(t \otimes t') \leq \underline{R}(t) \times \underline{R}(t')$ for any two tensors t and t'.

2.2 The exponent of matrix multiplication

The following theorem, which was obtained by Schönhage [13], shows that good upper bounds on ω can be obtained by finding a trilinear form of small border rank that can be degenerated into a direct sum of several large matrix multiplications.

Theorem 2.1. *Let e and m be two positive integers. Let t be a tensor such that $e \cdot \langle m, m, m \rangle \trianglelefteq t$. Then $em^\omega \leq \underline{R}(t)$.*

Our results will require a generalization of Theorem 2.1, based on the concept of *value* of a tensor. Our presentation of this concept follows [9]. Given a tensor $t \in \mathbb{F}^{u \times v \times w}$ and a positive integer N, define the set

$$\Big\{(e, m) \in \mathbb{N} \times \mathbb{N} \mid e \cdot \langle m, m, m \rangle \trianglelefteq (t \otimes t_{\mathsf{C}} \otimes t_{\mathsf{C}^2})^{\otimes N}\Big\} \qquad (2)$$

corresponding to all pairs (e, m) such that the tensor $(t \otimes t_{\mathsf{C}} \otimes t_{\mathsf{C}^2})^{\otimes N}$ can be degenerated into a direct sum of e tensors, each isomorphic to $\langle m, m, m \rangle$. Note that this set is finite. For any real number $\rho \in [2, 3]$, define

$$V_{\rho,N}(t) = \max\{(em^\rho)^{\frac{1}{3N}}\},$$

where the maximum is over all (e, m) in the set of Eq. (2). We now give the formal definition of the value of a tensor.

Definition 2.1. *For any tensor t and any $\rho \in [2, 3]$,*

$$V_\rho(t) = \lim_{N \to \infty} V_{\rho,N}(t).$$

The limit in this definition is well defined, see [9]. Obviously, $V_\rho(t) \geq V_\rho(t')$ for any tensors t, t' such that $t' \trianglelefteq t$. By definition, for any positive integers m, n and p we have $V_\rho(\langle m, n, p \rangle) \geq (mnp)^{\rho/3}$. Moreover, the value is superadditive and supermultiplicative: for any two tensors t and t', and any $\rho \in [2, 3]$, the inequalities $V_\rho(t \oplus t') \geq V_\rho(t) + V_\rho(t')$ and $V_\rho(t \otimes t') \geq V_\rho(t) \times V_\rho(t')$ hold. With this concept of value, we can state the following slight generalization of Theorem 2.1, which was used implicitly in [8] and stated explicitly in [9, 17].

Theorem 2.2. *Let t be a tensor and ρ be a real number such that $2 \leq \rho \leq 3$. If $V_\rho(t) \geq \underline{R}(t)$, then $\omega \leq \rho$.*

Finally, we will need the concept of decomposition of a tensor. Our presentation of this concept follows [5]. Let $t \in U \otimes V \otimes W$ be a tensor. Suppose that the vector spaces U, V and W decompose as

$$U = \bigoplus_{i \in I} U_i, \;\; V = \bigoplus_{j \in J} V_j, \;\; W = \bigoplus_{k \in K} V_k,$$

where I, J and K are three finite subsets of $\mathbb{Z}$. Let us call this decomposition D. We say that D is a decomposition of t if the tensor t can be written as

$$t = \sum_{(i,j,k) \in I \times J \times K} t(i, j, k),$$

where each $t(i, j, k)$ is a tensor in $U_i \otimes V_j \otimes W_k$ (the sum does not need to be direct). The support of t with respect to D is defined as

$$\mathrm{supp}(t) = \{(i, j, k) \in I \times J \times K \mid t(i, j, k) \neq 0\},$$

and the nonzero $t(i, j, k)$'s are called the components of t.

3. PRELIMINARIES AND NOTATIONS

In this section S denotes a finite subset of $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$.

Let $\alpha_1, \alpha_2, \alpha_3 \colon S \to \mathbb{Z}$ be the three coordinate functions of S, which means that $\alpha_\ell(\mathbf{s}) = s_\ell$ for each $\ell \in \{1, 2, 3\}$ and all $\mathbf{s} = (s_1, s_2, s_3) \in S$. We first define the concept of tightness. The same notion was used in [5].

Definition 3.1. *The set S is tight if there exists an integer d such that $\alpha_1(\mathbf{s}) + \alpha_2(\mathbf{s}) + \alpha_3(\mathbf{s}) = d$ for all $\mathbf{s} \in S$. The set S is b-tight, where b is a positive integer, if additionally $\alpha_\ell(S) \subseteq \{0, 1, \ldots, b-1\}$ for all $\ell \in \{1, 2, 3\}$.*

Note that if S is b-tight then $|S| \leq b^2$.

We denote by $\mathcal{F}(S)$ the set of all real-valued functions on S, and by $\mathcal{D}(S)$ the set of all probability distributions on S (i.e., the set of all functions $f \in \mathcal{F}(S)$ such that $f(\mathbf{s}) \geq 0$ for each $\mathbf{s} \in S$ and $\sum_{\mathbf{s} \in S} f(\mathbf{s}) = 1$). Note that, with pointwise addition and scalar multiplication, $\mathcal{F}(S)$ forms a real vector space of dimension $|S|$. Given any function $f \in \mathcal{F}(S)$, we denote by $f_1 \colon \alpha_1(S) \to \mathbb{R}$, $f_2 \colon \alpha_2(S) \to \mathbb{R}$ and $f_3 \colon \alpha_3(S) \to \mathbb{R}$ the three marginal functions of f: for each $\ell \in \{1, 2, 3\}$ and each $a \in \alpha_\ell(S)$,

$$f_\ell(a) = \sum_{\mathbf{s} \in \alpha_\ell^{-1}(a)} f(\mathbf{s}).$$

Let $\mathbb{S}_S$ denote the group of all permutations on S. Given any function $f \in \mathcal{F}(S)$ and any $\sigma \in \mathbb{S}_S$, we will denote by f^σ the function in $\mathcal{F}(S)$ such that $f^\sigma(\mathbf{s}) = f(\sigma(\mathbf{s}))$ for all $\mathbf{s} \in S$. We now define the concept of invariance of a function.

Definition 3.2. *Let G be a subgroup of $\mathbb{S}_S$. A function $f \in \mathcal{F}(S)$ is G-invariant if $f^\sigma = f$ for all $\sigma \in G$.*

We will denote by $\mathcal{F}(S, G)$ the set of all G-invariant real-valued functions on S, and by $\mathcal{D}(S, G) = \mathcal{D}(S) \cap \mathcal{F}(S, G)$ the set of all G-invariant probability distributions on S. We denote by $\mathcal{F}_0(S, G)$ the vector space of all functions $f \in \mathcal{F}(S, G)$ such that $f_\ell(a) = 0$ for all $\ell \in \{1, 2, 3\}$ and all $a \in \alpha_\ell(S)$, and write

$$\chi(S, G) = \dim(\mathcal{F}_0(S, G)).$$

We call this number $\chi(S, G)$ the compatibility degree of S with respect to G.

In our applications it will be sometimes more convenient to characterize the invariance in terms of a subgroup of permutations of the three coordinates of S, rather than in terms of a subgroup of permutations on S, as follows. Let L be a subgroup of $\mathbb{S}_3$, the group of permutations over $\{1, 2, 3\}$. We say that S is L-symmetric if $(s_{\sigma(1)}, s_{\sigma(2)}, s_{\sigma(3)}) \in S$ for all $(s_1, s_2, s_3) \in S$ and all $\sigma \in L$. If S is L-symmetric, the subgroup L induces the subgroup $L_S = \{\pi_\sigma \mid \sigma \in L\}$ of $\mathbb{S}_S$, where π_σ denotes the permutation in $\mathbb{S}_S$ such that $\pi_\sigma(s_1, s_2, s_3) = (s_{\sigma(1)}, s_{\sigma(2)}, s_{\sigma(3)})$ for all $(s_1, s_2, s_3) \in S$. We will slightly abuse notation and, when S is L-symmetric, simply write $\mathcal{F}(S, L)$, $\mathcal{D}(S, L)$, $\mathcal{F}_0(S, L)$, $\chi(S, L)$ to represent $\mathcal{F}(S, L_S)$, $\mathcal{D}(S, L_S)$, $\mathcal{F}_0(S, L_S)$ and $\chi(S, L_S)$, respectively.

The entropy of a probability distribution $P \in \mathcal{D}(S)$ is

$$H(P) = -\sum_{\mathbf{s}\in S} P(\mathbf{s}) \log(P(\mathbf{s})),$$

with the usual convention $0 \times \log(0) = 0$. Using the above notations, P_1, P_2 and P_3 represent the three marginal probability distributions of P. For each $\ell \in \{1, 2, 3\}$, the entropy of P_ℓ is

$$H(P_\ell) = -\sum_{a\in\alpha_\ell(S)} P_\ell(a) \log(P_\ell(a)).$$

It will sometimes be more convenient to represent, for each $\ell \in \{1, 2, 3\}$, the distribution P_ℓ as a vector $\mathbf{P}_\ell \in \mathbb{R}^{|\alpha_\ell(S)|}$, by fixing an arbitrary ordering of the elements in $\alpha_\ell(S)$.

We now define the concept of compatibility of two probability distributions.

Definition 3.3. *Two probability distributions P and Q in $\mathcal{D}(S)$ are compatible if $P_\ell = Q_\ell$ for each $\ell \in \{1, 2, 3\}$.*

Finally, for any $P \in \mathcal{D}(S)$, we define the quantity

$$\Gamma_S(P) = \max_Q [H(Q)] - H(P),$$

where the maximum is over all $Q \in \mathcal{D}(S)$ compatible with P. Note that $\Gamma_S(P)$ is always non-negative.

4. GENERAL THEORY

In this section we describe how to analyze the value of a trilinear form that has a decomposition with tight support.

4.1 Derivation of lower bounds on the value

Our main tool to analyze a trilinear form that has a decomposition with tight support is the following theorem, which shows how to reduce the computation of a lower bound on its value to solving an optimization problem. Due to space constraints, its proof is omitted from this version.

Theorem 4.1. *Let t be a trilinear form, and D be a decomposition of t with tight support $S \subseteq \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ and component set $\{t(\mathbf{s})\}_{\mathbf{s}\in S}$. Then, for any $P \in \mathcal{D}(S)$ and any $\rho \in [2, 3]$,*

$$\log(V_\rho(t)) \geq \sum_{\ell=1}^{3} \frac{H(P_\ell)}{3} + \sum_{\mathbf{s}\in S} P(\mathbf{s}) \log(V_\rho(t(\mathbf{s}))) - \Gamma_S(P).$$

This theorem can be seen as a generalized statement of the approach developed by Coppersmith and Winograd [8]. Several similar statements already appeared in the literature. A weaker statement, corresponding to the simpler case where each component is isomorphic to a matrix product (which removes the need for the term $-\Gamma_S(P)$ in the lower bound), can be found in [5]. The generalization to the case of arbitrary components stated in Theorem 4.1 was considered in [14], and proved implicitly, by considering several cases (symmetric and asymmetric supports) and without reference to the entropy, in [9, 12, 17]. Theorem 4.1 aims at providing a concise statement unifying all these results, described in terms of entropy in order to discuss the convexity of the lower bounds obtained. Ref. [11] also gives another exposition of this approach.

4.2 Solving the optimization problem

Let t be a trilinear form with a decomposition that has a tight support, as in the statement of Theorem 4.1. It will be convenient to define, for any $\rho \in [2, 3]$, the function $\Psi_{t,\rho} \colon \mathcal{D}(S) \to \mathbb{R}$ as

$$\Psi_{t,\rho}(P) = \sum_{\ell=1}^{3} \frac{H(P_\ell)}{3} + \sum_{\mathbf{s}\in S} P(\mathbf{s}) \log(V_\rho(t(\mathbf{s})))$$

for any $P \in \mathcal{D}(S)$. Note that this is a concave function on the convex set $\mathcal{D}(S)$. In order to optimize the lower bound on $\log(V_\rho(t))$ that is obtained from Theorem 4.1, we would like to find, for a given value of ρ, a probability distribution $P \in \mathcal{D}(S)$ that minimizes the expression

$$\Gamma_S(P) - \Psi_{t,\rho}(P).$$

This optimization problem is in general not convex, due to the presence of the term $\Gamma_S(P)$. In this subsection we develop a method to overcome this difficulty and find, using Theorem 4.1, a lower bound on $V_\rho(t)$ in polynomial time.

Remember that $\Gamma_S(P) = \max_Q[H(Q)] - H(P)$, where the maximum is over all $Q \in \mathcal{D}(S)$ that are compatible with P. When P is fixed, these conditions on Q can be written as linear constraints. Since the entropy is a strictly concave function, computing $-\Gamma_S(P)$ is then a strictly convex optimization problem on a convex set, and in particular has a unique solution $\hat{Q}$. Note that $\Gamma_S(\hat{Q}) = H(\hat{Q}) - H(\hat{Q}) = 0$, and thus $\Psi_{t,\rho}(\hat{Q})$ is a lower bound on $\log(V_\rho(t))$. The tightness of this lower bound of course depends on the initial choice of P. A natural choice is to take a probability distribution P that maximizes $\Psi_{t,\rho}(P)$, since finding such a probability distribution corresponds to solving a convex optimization problem. This motivates the algorithm described in Figure 1, which we call Algorithm $\mathcal{A}$.

```
Algorithm $\mathcal{A}$
Input: • the support $S \subseteq \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ of the tensor $t$
       • a value $\rho \in [2,3]$
       • the values $V_\rho(t(\mathbf{s}))$ for each $\mathbf{s} \in S$

1. Solve the following convex optimization problem.
       minimize    $-\Psi_{t,\rho}(P)$
       subject to  $P \in \mathcal{D}(S)$                 } OPT1

2. Solve the following convex optimization problem,
   where $\hat{P}$ denotes the solution found at Step 1.
       minimize    $-H(Q)$
       subject to  $Q \in \mathcal{D}(S)$                 } OPT2
                   $Q$ compatible with $\hat{P}$

3. Output $\Psi_{t,\rho}(\hat{Q})$, where $\hat{Q}$ denotes the solution
   found at Step 2.
```

Figure 1: Algorithm $\mathcal{A}$ computing, given $\rho \in [2, 3]$ and a tensor t with a decomposition that has a tight support, a lower bound on $\log(V_\rho(t))$.

As already mentioned, the optimization problem OPT 2 has a unique solution. While the solution of the optimization problem OPT1 may not be unique, it can actually be shown, using the strict concavity of the entropy function, that two solutions of OPT1 must have the same marginal probability distributions. Since the domain of the optimization problem OPT2 depends only on the marginal distributions of $\hat{P}$, the output of Algorithm $\mathcal{A}$ does not depend on which solution $\hat{P}$ was found at Step 1. This output is thus unique and, from Theorem 4.1 and the discussion above, it gives a lower bound on $\log(V_\rho(t))$. We state this conclusion in the following theorem.

Theorem 4.2. *If the support of t is tight, then Algorithm $\mathcal{A}$ outputs a lower bound on $\log(V_\rho(t))$.*

Note that the output of $\mathcal{A}$ is not always the best lower bound on $\log(V_\rho(t))$ that can be obtained from Theorem 4.1. This point will be discussed in the next subsection.

Let us now discuss the time complexity of implementing the algorithm of Figure 1. The worst-case running time depends on the time needed to solve the two optimization problems OPT1 and OPT2 at Steps 1 and 2. Let $v = \Psi_{t,\rho}(\hat{Q})$ denote the output of an exact implementation of Algorithm $\mathcal{A}$. Theorem 4.2 shows that $v \leq \log(V_\rho(t))$. Since both OPT1 and OPT2 are convex, and since the number of variables is upper bounded by $|S|$, for any $\varepsilon > 0$ both problems can be solved with accuracy ε in time $\mathrm{poly}(|S|, \log(1/\varepsilon))$ using standard methods [2, 10]. Thus, for any $\varepsilon' > 0$, we can compute in time $\mathrm{poly}(|S|, \log(1/\varepsilon'))$ a value v' such that $|v - v'| \leq \varepsilon' \cdot v$. In particular, we can use $\frac{v'}{1+\varepsilon'}$ as a lower bound on $\log(V_\rho(t))$.

We finally explain how to exploit symmetries of the decomposition of t to reduce the number of variables in Algorithm $\mathcal{A}$. These observations will enable us to slightly simplify the exposition of our results in the next sections. We first define invariance of a decomposition of a tensor.

Definition 4.1. *Let t be a tensor that has a decomposition D with support S and components $\{t(\mathbf{s})\}_{\mathbf{s}\in S}$. The decomposition D is G-invariant if $\Psi_{t,\rho}(P^\sigma) = \Psi_{t,\rho}(P)$ for any $P \in \mathcal{D}(S)$ and any $\sigma \in G$.*

With a slight abuse of language we will say, given a subgroup L of $\mathbb{S}_3$, that D is L-invariant if S is L-symmetric and D is L_S-invariant (see Section 3 for the definition of L_S).

Assume that the decomposition D of the tensor t on which we want to apply Algorithm $\mathcal{A}$ is G-invariant, where G is a subgroup of $\mathbb{S}_S$. Consider the optimization problem OPT1. Since the value of its objective function is then unchanged under the action of any permutation $\sigma \in G$ on P, OPT1 has a solution that is G-invariant (see, e.g., [4] for a discussion of symmetries in convex optimization). Now, if $\hat{P}$ is G-invariant, then the (unique) solution of the optimization problem OPT2 is G-invariant as well, since the value of the function $-H(Q)$ is unchanged under the action of any permutation on Q. This means that, if the decomposition D is G-invariant, then $\mathcal{D}(S)$ can be replaced by $\mathcal{D}(S,G)$ at both Steps 1 and 2 of Algorithm $\mathcal{A}$. Note that this set of probability distributions can be parametrized by $\dim(\mathcal{F}(S,G))$ parameters, instead of $|S|$ parameters.

4.3 Another approach

In this subsection we describe another approach to obtain lower bounds on $V_\rho(t)$ using Theorem 4.1, which is essentially how the powers of the construction by Coppersmith and Winograd were studied in previous works [9, 12, 14, 17]. Given any subgroup G of $\mathbb{S}_S$, let us consider the vector space $\mathcal{F}_0(S,G)$ of dimension $\chi(S,G)$ defined in Section 3. It will be convenient to represent functions in this vector space by vectors in $\mathbb{R}^{|S|}$, by fixing an arbitrary ordering of the elements in S. Let $\mathbf{R}$ be a generating matrix of size $|S| \times \chi(S,G)$ for $\mathcal{F}_0(S,G)$ (i.e., the columns of $\mathbf{R}$ form a basis of $\mathcal{F}_0(S,G)$). Since each coordinate of $\mathbb{R}^{|S|}$ corresponds to an element of S, we write $R_{\mathbf{s}j}$, for $\mathbf{s} \in S$ and $j \in \{1, \ldots, \chi(S,G)\}$, to represent the element in the $\mathbf{s}$-th row and the j-th column of $\mathbf{R}$. The approach is based on the following proposition, which is similar to a characterization given in [17]. Its proof is omitted from this version.

Proposition 4.1. *For any $P, P' \in \mathcal{D}(S,G)$ that are compatible, the equality $\Gamma_S(P') = H(P) - H(P')$ holds if P satisfies the following two conditions:*

(i) $P(\mathbf{s}) > 0$ for any $s \in S$ such that $\mathbf{R}$ contains at least one non-zero entry in its row labeled by $\mathbf{s}$,

(ii) $\sum_{\mathbf{s}\in S} R_{\mathbf{s}j} \log(P(\mathbf{s})) = 0$ for all $j \in \{1, \ldots, \chi(S,G)\}$.

In particular, applying Proposition 4.1 with $P' = P$ shows that, if Conditions (i) and (ii) are satisfied, then $\Gamma_S(P) = 0$, which implies $\log(V_\rho(t)) \geq \Psi_{t,\rho}(P)$ from Theorem 4.1. This motivates the algorithm described in Figure 2 that outputs a lower bound on $\log(V_\rho(t))$. We will call it Algorithm $\mathcal{B}$.

```
Algorithm $\mathcal{B}$

Input: • the support $S \subseteq \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ for the tensor $t$
       • a value $\rho \in [2,3]$
       • the values $V_\rho(t(\mathbf{s}))$ for each $\mathbf{s} \in S$
       • a subgroup $G$ of $\mathbb{S}_S$ such that the
             decomposition of $t$ is $G$-invariant

1. Solve the following optimization problem.
     minimize $-\Psi_{t,\rho}(P)$
     subject to $P \in \mathcal{D}(S,G)$
                $P$ satisfies Cond. (i)-(ii) of Prop. 4.1

2. Output $\Psi_{t,\rho}(\tilde{P})$, where $\tilde{P}$ denotes the solution
   found at Step 1.
```

Figure 2: Algorithm $\mathcal{B}$ computing, given $\rho \in [2,3]$ and a tensor t with a decomposition that has a tight support, a lower bound on $\log(V_\rho(t))$.

Note that, when $\chi(S,G) = 0$, Algorithms $\mathcal{A}$ and $\mathcal{B}$ solve exactly the same optimization problem (since in Algorithm $\mathcal{B}$ Conditions (i) and (ii) are satisfied for any $P \in D(S,G)$, and $\hat{Q} = \hat{P}$ in Algorithm $\mathcal{A}$) and thus output the same value. When $\chi(S,G) > 0$ Algorithm $\mathcal{B}$ usually gives better lower bounds than Algorithm $\mathcal{A}$, but at the price of introducing $\chi(S,G)$ highly nonconvex constraints, which makes the optimization problem much harder to solve, both in theory and in practice, even for a modest number of variables.

5. POWERS OF TENSORS

Let t and t' be two trilinear forms with decompositions D and D', respectively. Let $\mathrm{supp}(t) \subseteq \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ and $\mathrm{supp}(t') \subseteq$

$\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ denote their supports, and $\{t(\mathbf{s})\}_{\mathbf{s}\in\mathrm{supp}(t)}$ and $\{t'(\mathbf{s}')\}_{\mathbf{s}'\in\mathrm{supp}(t')}$ denote their component sets. Assume that both supports are tight. Fix $\rho\in[2,3]$ and assume that lower bounds on the values $V_\rho(t(\mathbf{s}))$ and $V_\rho(t'(\mathbf{s}'))$ are known for each $\mathbf{s}\in\mathrm{supp}(t)$ and each $\mathbf{s}'\in\mathrm{supp}(t')$. In this section we describe a method, inspired by [8] and [14], and also used in [17], to analyze $V_\rho(t\otimes t')$, and then show how to use it to analyze $V_\rho(t^{\otimes m})$ when m is a power of two.

In this section we will denote $\alpha_1,\alpha_2,\alpha_3\colon\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}\to\mathbb{Z}$ the three coordinate functions of $\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$.

Consider the tensor

$$t\otimes t'=\sum_{\mathbf{s}\in\mathrm{supp}(t)}\sum_{\mathbf{s}'\in\mathrm{supp}(t')}t(\mathbf{s})\otimes t'(\mathbf{s}').$$

Consider the following decomposition of $t\otimes t'$: the support $\mathrm{supp}(t\otimes t')\subseteq\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ is the set of all triples

$$(\alpha_1(\mathbf{s})+\alpha_1(\mathbf{s}'),\alpha_2(\mathbf{s})+\alpha_2(\mathbf{s}'),\alpha_3(\mathbf{s})+\alpha_3(\mathbf{s}'))$$

for $\mathbf{s}\in\mathrm{supp}(t)$ and $\mathbf{s}'\in\mathrm{supp}(t')$, and for each $(a,b,c)\in\mathrm{supp}(t\otimes t')$ the associated component is

$$(t\otimes t')(a,b,c)=\sum t(\mathbf{s})\otimes t'(\mathbf{s}'),$$

where the sum is over all $(\mathbf{s},\mathbf{s}')\in\mathrm{supp}(t)\times\mathrm{supp}(t')$ such that $\alpha_1(\mathbf{s})+\alpha_1(\mathbf{s}')=a$, $\alpha_2(\mathbf{s})+\alpha_2(\mathbf{s}')=b$ and $\alpha_3(\mathbf{s})+\alpha_3(\mathbf{s}')=c$. Note that the support of this decomposition is tight. If lower bounds on the value of each component are known, then we can use this decomposition to obtain a lower bound on $V_\rho(t\otimes t')$, by using Algorithm $\mathcal{A}$ on $t\otimes t'$, which requires solving two convex optimization problems, each having $|\mathrm{supp}(t\otimes t')|$ variables.

We now explain how to evaluate the value of those components $(t\otimes t')(a,b,c)$. For any $(a,b,c)\in\mathrm{supp}(t\otimes t')$, consider the following decomposition of $(t\otimes t')(a,b,c)$: the support is

$$\big\{\mathbf{s}\in\mathrm{supp}(t)\mid(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s}))\in\mathrm{supp}(t')\big\}$$

and, for each element $\mathbf{s}$ in this set, the corresponding component is

$$t(\mathbf{s})\otimes t'(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s})).$$

Note that the support in this decomposition is tight, and has size at most $|\mathrm{supp}(t)|$. The value of each component can be lower bounded as

$$\begin{aligned}V_\rho\left(t(\mathbf{s})\otimes t'(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s}))\right)\geq\\ V_\rho(t(\mathbf{s}))\times V_\rho(t'(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s})),\end{aligned}$$

from the supermultiplicativity of the value. As we supposed that the lower bounds on the values of each component of t and t' are known, we can use Algorithm $\mathcal{A}$ on each $(t\otimes t')(a,b,c)$ to obtain a lower bound on $V_\rho((t\otimes t')(a,b,c))$, which requires solving two convex optimization problems, each having at most $|\mathrm{supp}(t)|$ variables.

Let us now consider the case $t'=t$. We have just shown the following result: a lower bound on $V_\rho(t^{\otimes 2})$ can be computed by solving two convex optimization problems with $|\mathrm{supp}(t^{\otimes 2})|$ variables, and $2|\mathrm{supp}(t^{\otimes 2})|$ convex optimization problems with at most $|\mathrm{supp}(t)|$ variables. An important point is that this method additionally gives, as described in the previous paragraphs, a decomposition of $t^{\otimes 2}$ with tight support, and a lower bound on $V_\rho(t^{\otimes 2}(a,b,c))$ for each component $t^{\otimes 2}(a,b,c)$. This information can then be used to analyze the trilinear form $t^{\otimes 4}=t^{\otimes 2}\otimes t^{\otimes 2}$, by replacing t by $t^{\otimes 2}$ in the above analysis, giving a decomposition of $t^{\otimes 4}$ with tight support, a lower bound on $V_\rho(t^{\otimes 4})$ and a lower bound on the value $V_\rho(t^{\otimes 4}(a,b,c))$ of each component. By iterating this approach r times, for any $r\geq 1$, we can analyze the trilinear form $t^{\otimes 2^r}$, and in particular obtain a lower bound on $V_\rho(t^{\otimes 2^r})$. Let us denote by D^{2^r} the decomposition of $t^{\otimes 2^r}$ obtained by this approach. Its support $\mathrm{supp}(t^{\otimes 2^r})$ is the set of all triples

$$(\alpha_1(\mathbf{s})+\alpha_1(\mathbf{s}'),\alpha_2(\mathbf{s})+\alpha_2(\mathbf{s}'),\alpha_3(\mathbf{s})+\alpha_3(\mathbf{s}'))$$

for $\mathbf{s},\mathbf{s}'\in\mathrm{supp}(t^{\otimes 2^{r-1}})$. For any $(a,b,c)\in\mathrm{supp}(t^{\otimes 2^r})$, the corresponding component is

$$t^{\otimes 2^r}(a,b,c)=\sum t^{\otimes 2^{r-1}}(\mathbf{s})\otimes t^{\otimes 2^{r-1}}(\mathbf{s}'),$$

where the sum is over all $(\mathbf{s},\mathbf{s}')\in\mathrm{supp}(t^{\otimes 2^{r-1}})$ such that $\alpha_1(\mathbf{s})+\alpha_1(\mathbf{s}')=a$, $\alpha_2(\mathbf{s})+\alpha_2(\mathbf{s}')=b$ and $\alpha_3(\mathbf{s})+\alpha_3(\mathbf{s}')=c$. This approach also gives a decomposition $D^{2^r}_{abc}$ of each component $t^{\otimes 2^r}(a,b,c)$. In this decomposition the support, which we denote $S^{2^r}_{abc}$, is the set of all the elements $\mathbf{s}$ in $\mathrm{supp}(t^{\otimes 2^{r-1}})$ such that

$$(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s}))\in\mathrm{supp}(t^{\otimes 2^{r-1}}).$$

For any $\mathbf{s}\in S^{2^r}_{abc}$, the corresponding component of $t^{\otimes 2^r}(a,b,c)$ is

$$t^{\otimes 2^{r-1}}(\mathbf{s})\otimes t^{\otimes 2^{r-1}}(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s})).$$

The overall number of convex optimization problems that need to be solved in order to analyze $t^{\otimes 2^r}$ by the above approach is upper bounded by $r(2+2|\mathrm{supp}(t^{\otimes 2^r})|)$, while the number of variables in each optimization problem is upper bounded by $|\mathrm{supp}(t^{\otimes 2^r})|$. In the case where $\mathrm{supp}(t)$ is b-tight we can give a simple upper bound on this quantity. Indeed, when $\mathrm{supp}(t)$ is b-tight, our construction guarantees that $\mathrm{supp}(t^{\otimes 2^r})$ is $(b2^r)$-tight, which implies that $|\mathrm{supp}(t^{\otimes 2^r})|\leq(b2^r)^2$. We thus obtain the following result.

Theorem 5.1. *Let t be a trilinear form that has a decomposition with b-tight support* $\mathrm{supp}(t)$ *and components* $\{t(\mathbf{s})\}$. *Fix $\rho\in[2,3]$ and assume that a lower bound on the value $V_\rho(t(\mathbf{s}))$ is known for each $\mathbf{s}\in\mathrm{supp}(t)$. Then, for any integer $r\geq 1$, a lower bound on $V_\rho(t^{\otimes 2^r})$ can be computed by solving* $\mathrm{poly}(b,2^r)$ *convex optimizations problems, each optimization problem having* $\mathrm{poly}(b,2^r)$ *variables.*

We finally present two simple lemmas that show how to exploit the symmetries of the decomposition of t to reduce the number of variables. The proofs are omitted.

Lemma 5.1. *For any $r\geq 1$ and any $(a,b,c)\in\mathrm{supp}(t^{\otimes 2^r})$, the decomposition $D^{2^r}_{abc}$ is $\{\mathrm{id},\pi\}$-invariant, where* id *denotes the identity permutation and π is the permutation on $S^{2^r}_{abc}$ such that*

$$\pi(\mathbf{s})=(a-\alpha_1(\mathbf{s}),b-\alpha_2(\mathbf{s}),c-\alpha_3(\mathbf{s}))$$

for all $\mathbf{s}\in S^{2^r}_{abc}$.

Lemma 5.2. *Let L be a subgroup of $\mathbb{S}_3$. Assume that* $\mathrm{supp}(t)$ *is L-symmetric and that*

$$V_\rho(t(s_1,s_2,s_3))=V_\rho(t(s_{\sigma(1)},s_{\sigma(2)},s_{\sigma(3)}))$$

for any $\sigma\in L$ and any $\mathbf{s}=(s_1,s_2,s_3)\in\mathrm{supp}(t)$. Then D is L-invariant and, for any $r\geq 1$, the decomposition D^{2^r} is L-invariant as well.

From the discussion of Section 4.2, Lemma 5.1 enables us to reduce the number of variables when computing the lower bound on $V_\rho(t^{\otimes 2^r}(a,b,c))$ using Algorithm $\mathcal{A}$: instead of solving an optimization problem over $\mathcal{D}(\mathrm{supp}(t^{\otimes 2^r}(a,b,c)))$, we only need to consider $\mathcal{D}(\mathrm{supp}(t^{\otimes 2^r}(a,b,c)),\{\mathrm{id},\pi\})$. Similarly, if the conditions of Lemma 5.2 are satisfied, then, instead of considering $\mathcal{D}(\mathrm{supp}(t^{\otimes 2^r}))$, we need only to consider $\mathcal{D}(\mathrm{supp}(t^{\otimes 2^r}),L)$ when computing the lower bound on $V_\rho(t^{\otimes 2^r})$ using Algorithm $\mathcal{A}$.

Remark. The approach described in this section can be generalized to obtain lower bounds on $V_\rho(t^{\otimes m})$ when m is not a power of two. For instance the third power can be analyzed by studying $t \otimes t'$ with $t' = t^{\otimes 2}$. Another possible straightforward generalization is to allow other linear dependences in the definition of the support, i.e., defining the support of $t \otimes t'$ as the set of all triples

$$(\alpha_1(\mathbf{s}) + u\alpha_1(\mathbf{s}'), \alpha_2(\mathbf{s}) + u\alpha_2(\mathbf{s}'), \alpha_3(\mathbf{s}) + u\alpha_3(\mathbf{s}'))$$

for $\mathbf{s} \in \mathrm{supp}(t)$ and $\mathbf{s}' \in \mathrm{supp}(t')$, where $u \in \mathbb{Z}$ can be freely chosen. These two generalizations nevertheless do not seem to lead to any improvement for ω when applied to existing constructions.

6. APPLICATION

In this section we apply the theory developed in the previous sections to the construction $\mathfrak{t}$ by Coppersmith and Winograd, in order to obtain[2] upper bounds on ω.

6.1 Construction

Let $\mathbb{F}$ be an arbitrary field. Let q be a positive integer, and consider three vector spaces U, V and W of dimension $q+2$ over $\mathbb{F}$. Take a basis $\{x_0,\ldots,x_{q+1}\}$ of U, a basis $\{y_0,\ldots,y_{q+1}\}$ of V, and a basis $\{z_0,\ldots,z_{q+1}\}$ of W.

The trilinear form $\mathfrak{t}$ considered by Coppersmith and Winograd is the following trilinear form on (U,V,W):

$$\mathfrak{t} = \sum_{i=1}^{q}(x_0y_iz_i + x_iy_0z_i + x_iy_iz_0) + x_0y_0z_{q+1} + x_0y_{q+1}z_0 + x_{q+1}y_0z_0.$$

It was shown in [8] that $\underline{R}(\mathfrak{t}) = q+2$. Consider the following decomposition of U, V and W:

$$U = U_0 \oplus U_1 \oplus U_2,\ \ V = V_0 \oplus V_1 \oplus V_2,\ \ W = W_0 \oplus W_1 \oplus W_2,$$

where $U_0 = \mathrm{span}\{x_0\}$, $U_1 = \mathrm{span}\{x_1,\ldots,x_q\}$ and $U_2 = \mathrm{span}\{x_{q+1}\}$, $V_0 = \mathrm{span}\{y_0\}$, $V_1 = \mathrm{span}\{y_1,\ldots,y_q\}$ and $V_2 = \mathrm{span}\{y_{q+1}\}$, $W_0 = \mathrm{span}\{z_0\}$, $W_1 = \mathrm{span}\{z_1,\ldots,z_q\}$ and $W_2 = \mathrm{span}\{z_{q+1}\}$. This decomposition induces a decomposition D of $\mathfrak{t}$ with tight support

$$S = \{(2,0,0),(1,1,0),(1,0,1),(0,2,0),(0,1,1),(0,0,2)\}.$$

The components associated with $(2,0,0)$ and $(1,1,0)$ are

$$\mathfrak{t}(2,0,0) = x_{q+1}y_0z_0 \cong \langle 1,1,1\rangle,$$
$$\mathfrak{t}(1,1,0) = \sum_{i=1}^{q} x_iy_iz_0 \cong \langle 1,q,1\rangle.$$

We have $V_\rho(\mathfrak{t}(2,0,0)) = 1$ and $V_\rho(\mathfrak{t}(1,1,0)) \geq q^{\rho/3}$, from the definition of the value. The other components $\mathfrak{t}(0,2,0)$ and $\mathfrak{t}(0,0,2)$ are obtained by permuting the coordinates of $\mathfrak{t}(2,0,0)$, while the components $\mathfrak{t}(1,0,1)$ and $\mathfrak{t}(0,1,1)$ are obtained by permuting the coordinates of $\mathfrak{t}(1,1,0)$.

We now use Theorem 4.1 to obtain an upper bound on ω. Let P be a probability distribution in $\mathcal{D}(S)$. Let us write $P(2,0,0) = a_1$, $P(1,1,0) = a_2$, $P(1,0,1) = a_3$, $P(0,2,0) = a_4$, $P(0,1,1) = a_5$ and $P(0,0,2) = a_6$. The marginal distributions of P are $\mathbf{P}_1 = (a_1, a_2+a_3, a_4+a_5+a_6)$, $\mathbf{P}_2 = (a_4, a_2+a_5, a_1+a_3+a_6)$ and $\mathbf{P}_3 = (a_6, a_3+a_5, a_1+a_2+a_4)$. Since the only element in $D(S)$ compatible with P is P, we have $\Gamma_S(P) = 0$. Theorem 4.1 thus implies that

$$V_\rho(\mathfrak{t}) \geq \exp\left(\frac{H(P_1)+H(P_2)+H(P_3)}{3}\right) \times q^{(a_2+a_3+a_5)\rho/3}$$

for any $\rho \in [2,3]$. Evaluating this expression with $q = 6$, $a_2 = a_3 = a_5 = 0.3173$, $a_1 = a_4 = a_6 = (1-3a_2)/3$, and $\rho = 2.38719$ gives $V_\rho(\mathfrak{t}) > 8.00000017$. Using Theorem 2.2 and the fact that $\underline{R}(\mathfrak{t}) = q+2$, we conclude that $\omega < 2.38719$. This is the same upper bound as the bound obtained in Section 7 of [8].

6.2 Analyzing the powers using Algorithm $\mathcal{A}$

For any $r \geq 1$, we now consider the tensor $\mathfrak{t}^{\otimes 2^r}$ and analyze it using the framework and the notations of Section 5. The support of its decomposition D^{2^r} is the set of all triples

$$(a,b,c) \in \{0,\ldots,2^{r+1}\} \times \{0,\ldots,2^{r+1}\} \times \{0,\ldots,2^{r+1}\}$$

such that $a+b+c = 2^{r+1}$. Note that the decomposition D of $\mathfrak{t}$ satisfies the conditions of Lemma 5.2 for the subgroup $L = \mathbb{S}_3$ of $\mathbb{S}_3$, which implies that D^{2^r} is $\mathbb{S}_3$-invariant. Thus, from the discussion in Section 4.2, when applying Algorithm $\mathcal{A}$ on the trilinear form $t^{\otimes 2^r}$ in order to obtain a lower bound on $V_\rho(t^{2^r})$, we only need to consider probability distributions in $\mathcal{D}(\mathrm{supp}(\mathfrak{t}^{\otimes 2^r}),\mathbb{S}_3)$. This set can be parametrized by $\dim(\mathcal{F}(\mathrm{supp}(\mathfrak{t}^{\otimes 2^r}),\mathbb{S}_3))$ parameters. Remember that we also need a lower bound on the value of each component $\mathfrak{t}^{\otimes 2^r}(a,b,c)$ before applying $\mathcal{A}$ on $t^{\otimes 2^r}$. Using the method described in Section 5, these lower bounds are computed recursively by applying Algorithm $\mathcal{A}$ on the decomposition $D^{2^r}_{abc}$ of the component. Actually, we do not need to apply $\mathcal{A}$ when $a = 0$, $b = 0$ or $c = 0$, since a lower bound on the value can be found analytically in this case, as stated in the following lemma (see [17] for a proof).[3]

Lemma 6.1. *For any $r \geq 0$ and any $b \in \{0,1,\ldots,2^r\}$,*

$$V_\rho\big(\mathfrak{t}^{\otimes 2^r}(2^{r+1}-b,b,0)\big) \geq \left(\sum_{\substack{e\in\{0,\ldots,b\}\\ e\equiv b \bmod 2}} \frac{2^r!}{e!(\frac{b-e}{2})!(2^r-\frac{b+e}{2})!}q^e\right)^{\frac{\rho}{3}}.$$

Table 2 presents, for $r \in \{1,2,3,4,5\}$, the number of variables in the global optimization problem, the compatibility degree, and the best upper bound on ω we obtained by this approach. Computations have been done using the Matlab software CVX for convex optimization.

[2] All the programs used to perform the numerical calculations described in this section, and obtain our upper bounds on ω, are available as `http://www.francoislegall.com/MatrixMultiplication/programs.zip`.

[3] The lower bound of Lemma 6.1 is obtained directly (i.e., without using Theorem 4.1) by observing that, when $a = 0$, $b = 0$ or $c = 0$, the component is isomorphic to the tensor of a matrix product, and is actually better than the lower bound obtained by Algorithm $\mathcal{A}$ (or Algorithm $\mathcal{B}$).

Table 2: Analysis of $\mathfrak{t}^{\otimes 2^r}$ using Algorithm $\mathcal{A}$, with $q = 6$ for $r = 1$ and $q = 5$ for $r \in \{2, 3, 4, 5\}$.

r	dimension of $\mathcal{F}(\mathrm{supp}(\mathfrak{t}^{\otimes 2^r}), \mathbb{S}_3)$	$\chi(\mathrm{supp}(\mathfrak{t}^{\otimes 2^r}), \mathbb{S}_3)$	upper bound obtained
1	4	0	$\omega < 2.3754770$
2	10	2	$\omega < 2.3729372$
3	30	14	$\omega < 2.3728675$
4	102	70	$\omega < 2.3728672$
5	374	310	$\omega < 2.3728671$

6.3 Using both Algorithms $\mathcal{A}$ and $\mathcal{B}$

As mentioned in the introduction, the best known upper bound on ω obtained from the fourth power of $\mathfrak{t}$ is $\omega < 2.3729269$, which is slightly better than what we obtained in the previous subsection using Algorithm $\mathcal{A}$. This better bound can actually be obtained by using Algorithm $\mathcal{B}$ instead of Algorithm $\mathcal{A}$ when computing the lower bound on $V_\rho(\mathfrak{t}^{\otimes 4})$. More precisely, in this case the optimization problem in Algorithm $\mathcal{B}$ asks to minimize $-\Psi_{t,\rho}(P)$ such that $P \in \mathcal{D}(\mathrm{supp}(\mathfrak{t}^{\otimes 4}), \mathbb{S}_3)$ and P satisfies two additional constraints, since $\chi(\mathrm{supp}(\mathfrak{t}^{\otimes 4}), \mathbb{S}_3) = 2$. These constraints are highly non-convex but, since their number is only two, the resulting optimization problem can be solved fairly easily, giving the same upper bound $\omega < 2.3729269$.

We can also use Algorithm $\mathcal{B}$ instead of Algorithm $\mathcal{A}$ to analyze $\mathfrak{t}^{\otimes 8}$, but solving the corresponding optimization problems in this case was delicate and required a combination of several tools. We obtained lower bounds on the values of each component by solving the non-convex optimization problems using the NLPSolve function in Maple, while the lower bound on $V_\rho(\mathfrak{t}^{\otimes 8})$ has been obtained by solving the corresponding optimization problem (with 30 variables and 14 non-convex constraints) using the fmincon function in Matlab. The numerical calculations give

$$V_\rho(\mathfrak{t}^{\otimes 8}) > 5764802.8 > (q+2)^8,$$

which shows that $\omega < 2.3728642$.

While the non-convex optimization problems of Algorithm $\mathcal{B}$ seem intractable when studying higher powers of $\mathfrak{t}$, these powers can be analyzed by applying Algorithm $\mathcal{A}$, as in the previous subsection, but using this time the lower bounds on the values of the components $V_\rho(\mathfrak{t}^{\otimes 8}(a, b, c))$ obtained by Algorithm $\mathcal{B}$ as a starting point. This strategy can be equivalently described as using Algorithm $\mathcal{A}$ to analyze powers of $\mathfrak{t}'$, where $\mathfrak{t}' = \mathfrak{t}^{\otimes 8}$, with lower bounds on the values of each component of $\mathfrak{t}'$ computed by Algorithm $\mathcal{B}$. For $q = 5$ and $\rho = 2.3728640$, we obtain

$$V_\rho(\mathfrak{t}^{\otimes 16}) - (q+2)^{16} > 10^6,$$

which shows that $\omega < 2.3728640$. For $q = 5$ and $\rho = 2.3728639$, we obtain

$$V_\rho(\mathfrak{t}^{\otimes 32}) - (q+2)^{32} > 10^{20},$$

which shows that $\omega < 2.3728639$.

7. ACKNOWLEDGMENTS

The author is grateful to Virginia Vassilevska Williams for helpful correspondence, Yuval Filmus for comments about this paper, and to Harumichi Nishimura, Suguru Tamaki and Yuichi Yoshida for stimulating discussions. This work is supported by the Grant-in-Aid for Young Scientists (B) No. 24700005 of the JSPS and the Grant-in-Aid for Scientific Research on Innovative Areas No. 24106009 of the MEXT.

8. REFERENCES

[1] N. Alon, A. Shpilka, and C. Umans. On sunflowers and matrix multiplication. *Computational Complexity*, 22(2):219–243, 2013.

[2] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. SIAM, 2001.

[3] M. Bläser. *Fast Matrix Multiplication*. Number 5 in Graduate Surveys. Theory of Computing Library, 2013.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

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

[6] 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*, pages 379–388, 2005.

[7] H. Cohn and C. Umans. Fast matrix multiplication using coherent configurations. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1074–1087, 2013.

[8] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[9] A. M. Davie and A. J. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh*, 143A:351–370, 2013.

[10] S.-C. Fang and J. H.-S. Tsao. Entropy optimization: interior point methods. In *Encyclopedia of Optimization*, pages 544–548. Springer, 2001.

[11] Y. Filmus. Matrix multiplication I and II, 2014. Lecture notes available at `http://www.cs.toronto.edu/~yuvalf/`.

[12] F. Le Gall. Faster algorithms for rectangular matrix multiplication. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 514–523, 2012.

[13] A. Schönhage. Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3):434–455, 1981.

[14] A. Stothers. *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010.

[15] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[16] V. Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*, pages 49–54, 1986.

[17] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th ACM Symposium on Theory of Computing*, pages 887–898, 2012. Most recent version available at the author's homepage.