Fast Matrix Multiplication, the 3-tensor Rank Problem, and Moduli Spaces

Lars Hellström
lars.hellstrom@mdu.se

Mälardalen University

SNAG 2026
Linköping
2026-05-28

Abstract

Strassen's famous discovery of an $O(n^{2.81})$ algorithm for general matrix multiplication became a milestone in the theory of arithmetic complexity and sparked much research. It turns out the central object is the multiplication 3-tensor, known as the structure constants of an algebra, and the arithmetic complexity is determined by the rank of this tensor.
In principle the rank is algorithmically decidable, but since the trivial algorithm is an application of the Buchberger algorithm, in practice it is not. A confounding factor is that the objects considered exhibit a rather large symmetry group, which you normally would expect to make things easier, but when Gröbner bases are involved rather makes everything more difficult. Finding techniques to take advantage of symmetries is an important open problem, that perhaps SNAG could approach from new directions.
I will explain the problem basics and report on some (old) work by Daniel Andrén, Klas Markström, and myself on the matter.

Strassen's method

In 1969, Volker Strassen published the following algorithm for the 2×2 matrix product $C := AB$:

$$
\begin{aligned}
P_1 &:= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}); & P_5 &:= (A_{11} + A_{12}) \cdot B_{22}; \\
P_2 &:= (A_{21} + A_{22}) \cdot B_{11}; & P_6 &:= (A_{21} - A_{11}) \cdot (B_{11} + B_{12}); \\
P_3 &:= A_{11} \cdot (B_{12} - B_{22}); & P_7 &:= (A_{12} - A_{22}) \cdot (B_{21} + B_{22}); \\
P_4 &:= A_{22} \cdot (B_{21} - B_{11}); & & \\
C_{11} &:= P_1 + P_4 - P_5 + P_7; & C_{12} &:= P_3 + P_5; \\
C_{21} &:= P_2 + P_4; & C_{22} &:= P_1 - P_2 + P_3 + P_6;
\end{aligned}
$$

The important feature is that it performs only 7 multiplications (one for each P_i), as opposed to the 8 multiplications in the schoolbook algorithm.

Asymptotic complexity

Let $\mathrm{MM}(n)$ denote the minimal number of arithmetic operations needed to multiply two generic $n \times n$ matrices. Strassen observed that

$$\mathrm{MM}(2n) \leqslant 7\,\mathrm{MM}(n) + O(n^2)$$

because one way to multiply two $2n \times 2n$ matrices is to decompose them as 2×2 block matrices with $n \times n$ blocks, and then use the optimal $n \times n$ method to carry out the 7 multiplications on the previous slide. The rest of the arithmetic there is $O(n^2)$ operations to carry out for $n \times n$ blocks. Therefore

$$\mathrm{MM}(n) = O(n^{\log_2 7}) \approx O(n^{2.81})$$

which is an improvement on the $O(n^3)$ schoolbook algorithm.

More generally, any T multiplications method for multiplying $k \times k$ matrices implies

$$\mathrm{MM}(kn) \leqslant T\,\mathrm{MM}(n) + O(n^2),$$

yielding the bound $\mathrm{MM}(n) = O(n^{\log_k T})$.

General methods

A general such $n \times n$ matrix multiplication method would be defined by scalars $\{a_{ijk}\}_{i,j=1;k=1}^{n;T}$, $\{b_{rsk}\}_{r,s=1;k=1}^{n;T}$, and $\{c_{uvk}\}_{u,v=1;k=1}^{n;T}$ as

$$C_{uv} = \sum_{k=1}^{T} c_{uvk} P_k \quad \text{for } u, v \in \{1, \ldots, n\} \qquad \text{where}$$

$$P_k = \left(\sum_{i,j=1}^{n} a_{ijk} A_{ij} \right) \cdot \left(\sum_{r,s=1}^{n} b_{rsk} B_{rs} \right) \quad \text{for } k = 1, \ldots, T.$$

To give the correct result $C = AB$, it is necessary and sufficient that the scalars satisfy

$$\sum_{k=1}^{T} a_{ijk} b_{rsk} c_{uvk} = \begin{cases} 1 & \text{if } i = u,\ j = r, \text{ and } s = v, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i, j, r, s, u, v \in \{1, \ldots, n\}.$$

(Let A and B range over the matrices in the standard basis: $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$, and so on.)

Multiplication tensors

That right hand side

$$\gamma_{ijrs}^{uv} = \begin{cases} 1 & \text{if } i = u,\ j = r, \text{ and } s = v, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i, j, r, s, u, v \in \{1, \dots, n\}$$

is the $n \times n$ by $n \times n$ matrix multiplication tensor.
Indeed, for any N-dimensional algebra $(\mathcal{A}, \cdot)$ with basis $\{e_i\}_{i=1}^N$ there is a multiplication tensor γ whose defining identity is

$$\sum_{u=1}^{N} \gamma_{ir}^u e_u = e_i \cdot e_r \qquad \text{for } i, r \in \{1, \dots, N\}.$$

As an array of scalars, these γ_{ir}^u are also known as the structure constants of the algebra, but we *want* to view it as a tensor.

As a straight up tensor, $\gamma \in \mathcal{A} \otimes \mathcal{A}^* \otimes \mathcal{A}^*$ or $\gamma \colon \mathcal{A} \otimes \mathcal{A} \longrightarrow \mathcal{A}$, but it is common to recast factors between algebra $\mathcal{A}$ and dual $\mathcal{A}^*$.

Tensor decomposition

Let $\mathbb{F}$ be some field and V an N-dimensional vector space over $\mathbb{F}$.

In the coordinate setting, those "general methods" for a multiplication $\gamma\colon V \otimes V \longrightarrow V$ are decompositions

$$\gamma_{ir}^{u} = \sum_{k=1}^{T} c_{uk} a_{ik} b_{rk} \qquad \text{for } i, r, u \in \{1, \dots, N\}.$$

The coordinate-free formulation of this is that

$$\gamma = \sum_{k=1}^{T} \mathbf{c}_k \otimes \mathbf{a}_k \otimes \mathbf{b}_k$$

for some $\{\mathbf{c}_k\}_{k=1}^{T} \subseteq V$ and $\{\mathbf{a}_k\}_{k=1}^{T}, \{\mathbf{b}_k\}_{k=1}^{T} \subseteq V^*$.

Here, we write a general tensor γ as a sum of simple tensors $\mathbf{c}_k \otimes \mathbf{a}_k \otimes \mathbf{b}_k$. The smallest T for which this is possible is called the rank of the tensor γ.

Complex multiplication over reals

A classical trick is that for $x, y, u, v \in \mathbb{R}$,

$$(x+iy)\cdot(u+iv) = (p_1 - p_3) + i(p_2 - p_1 - p_3) \qquad \text{where} \quad \begin{cases} p_1 = x \cdot u, \\ p_2 = (x+y)\cdot(u+v), \\ p_3 = y \cdot v. \end{cases}$$

This translates into the decomposition

$$\gamma_{\mathbb{C}/\mathbb{R}} = (1-i) \otimes \mathrm{Re} \otimes \mathrm{Re} + i \otimes (\mathrm{Re} + \mathrm{Im}) \otimes (\mathrm{Re} + \mathrm{Im}) + (-1-i) \otimes \mathrm{Im} \otimes \mathrm{Im}$$

of the tensor $\gamma_{\mathbb{C}/\mathbb{R}} \colon \mathbb{C} \otimes_{\mathbb{R}} \mathbb{C} \longrightarrow \mathbb{C}$ for complex multiplication over the reals.

This demonstrates that the rank of $\gamma_{\mathbb{C}/\mathbb{R}}$ is $\leqslant 3$, not 4 as the schoolbook decomposition

$$\gamma_{\mathbb{C}/\mathbb{R}} = 1 \otimes \mathrm{Re} \otimes \mathrm{Re} + i \otimes \mathrm{Im} \otimes \mathrm{Re} + i \otimes \mathrm{Re} \otimes \mathrm{Im} - 1 \otimes \mathrm{Im} \otimes \mathrm{Im}$$

would suggest.

2-tensor rank = matrix rank

The reason to use 'rank' as name for this concept is that it generalises 'rank of matrix'; tensors with 2 indices are matrices. Concretely,

$$(\text{rank of } M \in \mathbb{F}^{n\times n}) = \big(\text{smallest } T \text{ such that } M = \textstyle\sum_{k=1}^{T} \mathbf{a}_k \mathbf{b}_k^{\mathrm{T}} \text{ for some } \{\mathbf{a}_k\}_{k=1}^{T}, \{\mathbf{b}_k\}_{k=1}^{T} \subseteq \mathbb{F}^n\big).$$

Why? Any M has an LU factorisation: $PM = LU$ where P is a permutation matrix, L is lower triangular, and U is on row-echelon form; the matrix rank of M is the number of nonzero rows of U. For T equal to that rank, and $\{\mathbf{e}_i\}_{i=1}^{n}$ denoting the standard basis in $\mathbb{F}^n$, we have

$$M = P^{-1}LU = P^{-1}L\left(\sum_{k=1}^{T} \mathbf{e}_k \mathbf{e}_k^{\mathrm{T}}\right)U = \sum_{k=1}^{T} P^{-1}L\mathbf{e}_k \mathbf{e}_k^{\mathrm{T}}U = \sum_{k=1}^{T} \mathbf{a}_k \mathbf{b}_k^{\mathrm{T}}$$

for $\mathbf{a}_k = P^{-1}L\mathbf{e}_k$ and $\mathbf{b}_k = U^{\mathrm{T}}\mathbf{e}_k$.

3-tensor rank is *hard*

In principle, the rank of a 3-tensor is decidable. For example, to prove that the rank of the 2×2 matrix multiplication tensor is > 6, it suffices to check that the equation system

$$\sum_{k=1}^{6} a_{ijk} b_{rsk} c_{uvk} = \begin{cases} 1 & \text{if } i = u,\ j = r, \text{ and } s = v, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i, j, r, s, u, v \in \{1, 2\}$$

has no solutions. Over an algebraically closed field, that is equivalent to the ideal membership problem

$$1 \in \left\langle \sum_{k=1}^{6} a_{ijk} b_{rsk} c_{uvk} - \begin{cases} 1 & \text{if } i = u,\ j = r, \text{ and } s = v, \\ 0 & \text{otherwise} \end{cases} \right\rangle_{i,j,r,s,u,v \in \{1,2\}}$$

which is algorithmically solvable using Gröbner bases; this is all *commutative* algebra.

The catch is that we have $3 \cdot 2 \cdot 2 \cdot 6 = 72$ variables. The Buchberger algorithm is doubly exponential in the number of variables; $2^{2^{72}}$ is too much work.

3-tensor rank is NP-hard

Johan Håstad (KTH) proved (1990) that determining the rank of a 3-tensor is NP-hard, by direct encoding:

Theorem

From an instance of 3SAT with n variables and m clauses, one may construct a $(2+n+2m) \times 3n \times (3n+m)$ tensor that has rank $4n+2m$ if and only if the instance is satisfiable; otherwise the rank is larger.

This is a rather large tensor, and high rank, so if this was where things get tough then we should not worry. But tensor rank gets difficult much sooner than that.

For comparison, the 3×3 matrix multiplication tensor is size $9 \times 9 \times 9$. The best known bounds for its rank T are

$$19 \leqslant T \leqslant 23.$$

The SAT approach

For any *finite* field $\mathbb{F}$, the equation system in the scalars a_{ijk}, b_{rsk}, and c_{uvk} can conversely be encoded as an instance of boolean satisfiability (SAT).
In particular, any *integer coefficient* solution would give rise to a solution modulo 2, which shows up over the binary field $\mathbb{F}_2$.

Around the turn of the millenium, SAT-solver technology made major strides — in particular the invention of Conflict-Driven Clause Learning solvers — and as a result it became feasible to *search* the space of decompositions.
At least for small sizes.

Even so, performance of SAT-solvers is *way* better than that of Gröbner basis techniques.
The memory footprint in particular is much more manageable.

The XOR problem

The standard CNF encoding of a SAT problem is a list of *clauses*, such as

$$x_1 \vee x_2 \vee \neg x_4 \vee x_7$$

Each clause forbids *one* assignment of values to the variables it mentions (here $x_1 = 0$, $x_2 = 0$, $x_4 = 1$, and $x_7 = 0$ will not satisfy that clause).

This is universal: any constraint can be expressed in clauses.

Some constraints are more cumbersome than others, however. XOR of things is awkward, because changing any input to a XOR changes the output.
Addition in $\mathbb{F}_2$ is exactly XOR. Our sums $\sum_{k=1}^{T} a_k b_k c_k$ have a lot of terms.
This can be made manageable by introducing helper variables for the values of partial sums.

It's still not easy

Only rephrasing as SAT is not enough.

Modern SAT-solvers are quite good at eliminating 'corners' of the search space. But each symmetry of the problem creates mirror images of every imaginable corner, so the solver keeps encountering disguised duplicates of subproblems it has already solved.

The decomposition of the matrix multiplication tensor problem has *tons* of symmetries.

Continuous symmetries

For any $U, V, W \in \mathrm{GL}(n, \mathbb{F})$, we have

$$C = A \cdot B \quad \Longleftrightarrow \quad UCW^{-1} = UAV^{-1} \cdot VBW^{-1}.$$

This translates to actions of U, V, W on the scalar unknowns $a_{ijk}, b_{rsk}, c_{uvk}$.

Write-up conundrum

That makes for a $\mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$ symmetry group, with (close to) $3n^2$ degrees of freedom.

A general multiplication tensor on an N-dimensional algebra has a $\mathrm{GL}(N, \mathbb{F})$ symmetry group from changes of basis.
But $\mathbb{F}^{n \times n}$ is n^2-dimensional, so that suggests there should be a continuous symmetry group with $(n^2)^2 = n^4 > 3n^2$ degrees of freedom.

Discrete symmetries

The transpose matrix identity

$$(AB)^{\mathrm{T}} = B^{\mathrm{T}} A^{\mathrm{T}}$$

gives rise to a symmetry turning every a_{ijk} into a b_{rsk}, and vice versa.

In fact you can freely permute the roles of A, B, and C in the operation: any decomposition of the $l \times m$ by $m \times n$ to $l \times n$ matrix multiplication tensor also gives a decomposition of the $m \times n$ by $n \times l$ to $m \times l$ matrix multiplication tensor. This is not intuitive, but obvious when you compare the equation systems.

Thus there is a symmetric group S_3 acting on $\mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$.

But the *big* discrete symmetry comes from permuting the terms in the tensor decomposition: this is a whopping S_T with $T!$ elements.
That gives us a combined symmetry group of

$$\big(\mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})\big) \rtimes S_3 \times S_T.$$

Symmetry breaking

In the SAT setting, the way to make use of symmetries is to *break* them: we only want to look at one point from each orbit of the symmetry group.

Practically, we can collect all our variables $a_{ijk}, b_{rsk}, c_{uvk}$ as elements of one vector $\mathbf{x}$.
Any symmetry σ acts on $\mathbf{x}$ to produce a new vector $\sigma(\mathbf{x})$.
We can single out one point from each orbit by imposing the additional constraints that

$$\mathbf{x} \leqslant \sigma(\mathbf{x}) \qquad \text{for all symmetries } \sigma,$$

where $\leqslant$ is any total ordering of the set of such vectors; we used a simple lexicographic order.

Rumor has it the late 1990's Per Ling (then Umeå University CS department) ran several workstations for two months to search *part* of the space of rank 7 decompositions of 2×2 matrix multiplication over $\mathbb{F}_2$.
In 2003, using SAT-solving and symmetry breaking, we were able to search *all* of that space in five minutes, on a laptop.

Symmetry breaking details

The FMM symmetry group G is too large to include $\mathbf{x} \leqslant \sigma(\mathbf{x})$ for all $\sigma \in G$.

The symmetric group S_T is *nice*, in that it suffices to let σ range over the transpositions $(k\ k{+}1)$; the rest then follows from transitivity of $\leqslant$.

Of the $\mathrm{GL}(n, \mathbb{F})$ groups we only ever used the permutation matrices, so a few more S_n's.
We considered including inequalities from the other elementary row operations as well, but never got around to implementing that.
Over $\mathbb{F}_2$, there aren't that many such operations, and IIRC several would yield nothing new.

For cartesian products of groups, we did range over combinations of generator for one factor and generator for the other.

Daniel Andrén points out that *order of variables* in the vector $\mathbf{x}$ was a parameter we did not explore. SAT benefits from having good locality of the constraints, i.e., many speak about the same subset of variables.

Algebra–geometry phrasebook

Let $J \subseteq \mathbb{F}[X]$ be an ideal and consider the corresponding variety $\mathcal{V}(J) \subseteq \mathbb{F}^{|X|}$.
If $\mathcal{V}(J)$ has a group G of symmetries, then modding out those means moving to the quotient $\mathcal{V}(J)/G$.
On the algebra side, the counterpart of taking the quotient is supposed to be to work in the invariant ring $\mathbb{F}[X]^G$.

At the time, I couldn't find any readable explanation of how to do that.

Part of the problem may be that $\mathbb{F}[X]^G$ is a subalgebra, and e.g. SAGBI bases are hairier than Gröbner bases.
We'd need to *then* consider ideals in this subalgebra!

Might it be easier to deal with such things noncommutatively, as ideals in some $\mathbb{F}[X] \rtimes G$?

Gröbner modulo symmetries?

If $\mathcal{Y}$ is the monoid of power products in $\mathbb{F}[X]$, and G is a finite group acting on $\mathbb{F}[X]$, then $\mathbb{F}[X] \rtimes G$ has a vector space basis like $\mathcal{Y} \times G$, which is larger than that of $\mathbb{F}[X]$.
Moreover $\mathbb{F}[X] \rtimes G$ is noncommutative!

On the other hand, in $\mathbb{F}[X] \rtimes G$ there are more ways that a monomial can be a multiple of another than there is in $\mathbb{F}[X]$—having some $\sigma \in G$ act on a monomial $\mu \in \mathcal{Y}$ is a multiple

$$\sigma(\mu) = \sigma \cdot \mu \cdot \sigma^{-1}.$$

Thus there are *fewer* monoidal ideals in $\mathbb{F}[X] \rtimes G$ than in $\mathbb{F}[X]$.

A complication is that a total ordering cannot be strictly compatible with multiplication by elements in a finite group, thus it can happen that

$$\mathrm{lm}(\rho \cdot b \cdot \sigma) \neq \rho \cdot \mathrm{lm}(b) \cdot \sigma;$$

taking multiples may change which term of a polynomial b is the leading term.
This would require reworking the *formalism* of Gröbner bases.

How far we got

We exhausted 2×2 (rank 7) and 2×2 by 2×3 (rank 11) matrix multiplication. We were not able do 3×2 by 2×3 (rank 15) matrix multiplication, which had anyway been solved theoretically long ago.

We did cover a number of submatrix problems: tensors for multiplying matrices where some elements are fixed at 0.

It turns out there is a general theory where such methods are useful, at least for improving the asymptotic exponent.
The trick is to have a method that computes two different tensors of the same arguments, in fewer multiplications that would be needed to compute both separately.
Then several such methods are fit together, jigsaw style.

Subproblem rank constraints

Our idea was rather to constrain the *sparseness* of decompositions.

In e.g. the 3×3 multiplication tensor, there are a number of 2×2 multiplication tensors obtained by deleting some row and column.
Formally, those can be described as projections of the 3×3 multiplication tensor.
At least 7 terms in any decomposition of the 3×3 tensor have to be nonzero under those projections, because otherwise it would do 2×2 multiplication in less than 7 products.

For any subproblem of known rank, this imposes a highly nontrivial bound on the zeroes in a decomposition.
But these new constraints were not sufficient to solve any new instances.

Other applications

3-tensor decomposition is not the only problem that can be encoded using SAT. Any problem with a finite search space could be a candidate.

One not-too-different problem would be that of enumerating finite-dimensional algebras of a particular kind.
In this case, the primary unknowns are the "structure constants". The equations come from the identities you wish to impose.
We again have a large symmetry group from changes of basis.

Main catch is that you get algebras over $\mathbb{F}_2$.
Other finite fields (or rings) are equally *possible*, but more cumbersome.

That's all

Thank you for listening.