Brief Announcement: Tight Memory-Independent Parallel Matrix Multiplication Communication Lower Bounds

Hussam Al Daas
Rutherford Appleton Laboratory
Didcot, Oxfordshire, UK
hussam.al-daas@stfc.ac.uk

Grey Ballard
Wake Forest University
Winston-Salem, NC, USA
ballard@wfu.edu

Laura Grigori
Inria Paris
Paris, France
laura.grigori@inria.fr

Suraj Kumar
Inria Paris
Paris, France
suraj.kumar@inria.fr

Kathryn Rouse
Inmar Intelligence
Winston-Salem, NC, USA
kathryn.rouse@inmar.com

ABSTRACT

Communication lower bounds have long been established for matrix multiplication algorithms. However, most methods of asymptotic analysis have either ignored constant factors or not obtained the tightest possible values. The main result of this work is establishing memory-independent communication lower bounds with tight constants for parallel matrix multiplication. Our constants improve on previous work in each of three cases that depend on the relative sizes of the matrix aspect ratios and the number of processors.

CCS CONCEPTS

• **Theory of computation → Massively parallel algorithms**.

KEYWORDS

Rectangular matrix multiplication, convex optimization

ACM Reference Format:
Hussam Al Daas, Grey Ballard, Laura Grigori, Suraj Kumar, and Kathryn Rouse. 2022. Brief Announcement: Tight Memory-Independent Parallel Matrix Multiplication Communication Lower Bounds. In *SPAA '22: ACM Symposium on Parallelism in Algorithms and Architectures, July 11–14, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3490148.3538552

1 INTRODUCTION

The cost of communication relative to computation continues to grow, so the time complexity of an algorithm must account for both the computation it performs and the data that it communicates. Communication lower bounds for computations set targets for efficient algorithms and spur algorithmic development. Classical matrix multiplication is one of the most fundamental computations, and its I/O complexity on sequential machines and parallel communication costs have been well studied over decades [1, 7, 8].

The earliest results established asymptotic lower bounds, ignoring constant factors and lower order terms. Because of the ubiquity of matrix multiplication in high performance computations, more recent attempts have tightened the analysis to obtain the best constant factors for memory-dependent sequential bounds [10]. These improvements in the lower bound also helped identify the best performing algorithms to be further tuned for high performance in settings where small constant factors make significant differences.

The main result of this paper is the establishment of tight constants for memory-independent communication lower bounds for parallel matrix multiplication. These bounds apply even when the local memory is infinite, and they are the tightest bounds in many cases when the memory is limited. Demmel et al. [6] prove asymptotic bounds for general rectangular matrix multiplication and show that three different bounds are asymptotically tight in separate cases that depend on the relative sizes of the aspect ratios of the matrices and the number of processors. Our main result reproduces those asymptotic bounds and improves the constants in all three cases. We present a comparison to previous work in Tab. 1.

We believe one of the main features of our lower bound result is the simplicity of the proof technique, which makes a unified argument that applies to all three cases. The key idea is to cast the lower bound as the solution to a constrained optimization problem whose objective function is the sum of variables that correspond to the amount of data of each matrix required by a single processor's computation. All of the complexity of the three cases, including establishing the thresholds between cases and the leading terms in each case, are confined to a single optimization problem that can be solved analytically. This unified argument is elegant and improves on previous results to obtain tight constants.

2 PRELIMINARIES

2.1 Parallel Computation Model

We consider the α-β-γ parallel machine model [11]. In this model, each of P processors has its own local memory of size M and can compute only with data in its local memory. The processors can communicate data to and from other processors via messages that are sent over a fully connected network. The cost of communication is a function of two parameters α and β, where α is the per-message latency cost and β is the per-word bandwidth cost, and γ is the per-operation arithmetic cost. We focus on the bandwidth cost in this work, as it typically dominates the communication cost for dense matrix multiplication.

SPAA '22, July 11–14, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9146-7/22/07.
https://doi.org/10.1145/3490148.3538552

	$1 \le P \le \frac{m}{n}$	$\frac{m}{n} \le P \le \frac{mn}{k^2}$	$\frac{mn}{k^2} \le P$
Leading term	nk	$\left(\frac{mnk^2}{P}\right)^{1/2}$	$\left(\frac{mnk}{P}\right)^{2/3}$
[1]	-	-	$\left(\frac{1}{2}\right)^{2/3} \approx .63$
[8]	-	-	$\frac{1}{2} = .5$
[6]	$\frac{16}{25} = .64$	$\left(\frac{2}{3}\right)^{1/2} \approx .82$	1
Theorem 1	1	2	3

Table 1: Summary of explicit constants of leading term of parallel memory-independent rectangular matrix multiplication communication lower bounds for multiplication dimensions $m \ge n \ge k$ and P processors

2.2 Related Work

Aggarwal, Chandra, and Snir [1] extend the sequential lower bound for matrix multiplication of Hong and Kung [7] to the LPRAM parallel model, which closely resembles the model we consider with the exception that there exists a global shared memory where the inputs initially reside and where the output must be stored at the end of the computation. In addition to proving bounds for sequential matrix multiplication and an associated memory-dependent bound for parallel matrix multiplication, Irony, Toledo, and Tiskin [8] prove also that a parallel algorithm must communicate $\Omega(n^2/P^{2/3})$ words, and they provide explicit constants in their analysis. Demmel et al. [6] extend the memory-independent results to the rectangular case (multiplying matrices of dimensions $n_1 \times n_2$ and $n_2 \times n_3$), showing that three different bounds apply that depend on the relative sizes of the three dimensions and the number of processors, and their proof provides explicit constants. We summarize the constants obtained by these previous works and compare them to our results in Tab. 1. Further details of the comparison are given in [2].

2.3 Fundamental Results

In this section we collect the fundamental existing results we use to prove our main result, Theorem 1. The first lemma is a geometric inequality that has been used before in establishing communication lower bounds for matrix multiplication [3, 6, 8].

Lemma 1 (Loomis-Whitney [9]). *Let V be a finite set of lattice points in $\mathbb{R}^3$, i.e., points (i, j, k) with integer coordinates. Let $\phi_i(V)$ be the projection of V in the i-direction, i.e., all points (j, k) such that there exists an i so that $(i, j, k) \in V$. Define $\phi_j(V)$ and $\phi_k(V)$ similarly. Then*

$$|V| \le |\phi_i(V)| \cdot |\phi_j(V)| \cdot |\phi_k(V)|,$$

where $|\cdot|$ denotes the cardinality of a set.

The next set of definitions and lemmas allow us to solve the key constrained optimization problem (Lemma 5) analytically. We use boldface to indicate vectors and matrices and subscripts to index them, so that x_i is the ith element of $\boldsymbol{x}$, for example.

Definition 1 ([5, eq. (3.2)]). *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is* convex *if its domain is a convex set and for all $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}\ f$,*

$$f(\boldsymbol{y}) \ge f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle.$$

Definition 2 ([5, eq. (3.20)]). *A differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is* quasiconvex *if its domain is a convex set and for all $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}\ g$,*

$$g(\boldsymbol{y}) \le g(\boldsymbol{x}) \text{ implies that } \langle \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \le 0.$$

Definition 3 ([5, eq. (5.49)]). *Consider an optimization problem of the form*

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0} \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ and $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^c$ are both differentiable. Define the dual variables $\boldsymbol{\mu} \in \mathbb{R}^c$, and let $\boldsymbol{J_g}$ be the Jacobian of $\boldsymbol{g}$. The Karush-Kuhn-Tucker (KKT) *conditions of $(\boldsymbol{x}, \boldsymbol{\mu})$ are as follows:*

- Primal feasibility*: $\boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}$;*
- Dual feasibility*: $\boldsymbol{\mu} \ge 0$;*
- Stationarity*: $\nabla f(\boldsymbol{x}) + \boldsymbol{\mu} \cdot \boldsymbol{J_g}(\boldsymbol{x}) = \boldsymbol{0}$;*
- Complementary slackness*: $\mu_i g_i(\boldsymbol{x}) = 0$ for all $i \in \{1, \ldots, c\}$.*

The next two results establish that our key optimization problem in Lemma 5 can be solved analytically using the KKT conditions. Proofs of these results can be found in [2].

Lemma 2 ([4, Lemma 2.2]). *The function $g_0(\boldsymbol{x}) = L - x_1 x_2 x_3$, for some constant L, is quasiconvex in the positive octant.*

Lemma 3. *Consider an optimization problem of the form given in eq. (1). If f is a convex function and each g_i is a quasiconvex function, then the KKT conditions are sufficient for optimality.*

3 MAIN LOWER BOUND RESULT

3.1 Lower Bounds on Individual Array Access

The following lemma establishes lower bounds on the number of elements of each individual matrix a processor must access based on the number of computations a given element is involved with. This result is used to establish a set of constraints in the key optimization problem used in the proof of Theorem 1.

Lemma 4. *Given a parallel matrix multiplication algorithm that multiplies an $n_1 \times n_2$ matrix $\mathbf{A}$ by an $n_2 \times n_3$ matrix $\mathbf{B}$ using P processors, any processor that performs at least $1/P$th of the scalar multiplications must access at least $n_1 n_2/P$ elements of $\mathbf{A}$ and at least $n_2 n_3/P$ elements of $\mathbf{B}$ and also compute contributions to at least $n_2 n_3/P$ elements of $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$.*

Proof. The total number of scalar multiplications that must be computed is $n_1 n_2 n_3$. Consider a processor that computes at least $1/P$th of these computations. Each element of $\mathbf{A}$ is involved in n_3 multiplications. If the processor accesses fewer than $n_1 n_2/P$ elements of $\mathbf{A}$, then it would perform fewer than $n_3 \cdot n_1 n_2/P$ scalar multiplications, which is a contradiction. Likewise, each element of $\mathbf{B}$ is involved in n_1 multiplications. If the processor accesses fewer than $n_2 n_3/P$ elements of $\mathbf{B}$, then it would perform fewer than $n_1 \cdot n_2 n_3/P$ scalar multiplications, which is a contradiction. Finally, each element of $\mathbf{C}$ is the sum of n_2 scalar multiplications. If the processor computes contributions to fewer than $n_1 n_3/P$ elements of $\mathbf{C}$, then it would perform fewer than $n_2 \cdot n_1 n_3/P$ scalar multiplications, which is again a contradiction. □

3.2 Key Optimization Problem

The following lemma is the crux of the proof of our main result (Theorem 1). We state the optimization problem abstractly here, but it may be useful to have the following intuition: the variable vector $\boldsymbol{x}$ represents the sizes of the projections of the computation assigned to a single processor onto the three matrices, where x_1 corresponds to the smallest matrix and x_3 corresponds to the largest matrix. In order to design a communication-efficient algorithm, we wish to minimize the sum of the sizes of these projections subject to the constraints of matrix multiplication (and the processor performing $1/P$th of the computation), as specified by the Loomis-Whitney inequality (Lemma 1) and Lemma 4. A more rigorous argument that any parallel matrix multiplication algorithm is subject to these constraints is given in Theorem 1.

We are able to solve this optimization problem analytically using properties of convex optimization (Lemma 3). The three cases of the solution correspond to how many of the individual variable constraints are tight. When none of them is tight, we can minimize the sum of variables subject to the bound on their product by setting them all equal to each other (Case 3). However, when the individual variable constraints make this solution infeasible, those become active and the free variables must be adjusted (Cases 1 and 2).

Lemma 5. *Consider the following optimization problem:*

$$\min_{\boldsymbol{x}\in\mathbb{R}^3} x_1 + x_2 + x_3$$

such that

$$\left(\frac{mnk}{P}\right)^2 \le x_1x_2x_3, \quad \frac{nk}{P} \le x_1, \quad \frac{mk}{P} \le x_2, \quad \frac{mn}{P} \le x_3,$$

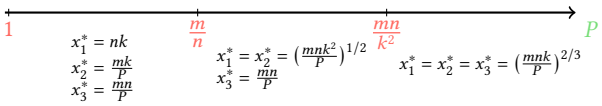
where $m \ge n \ge k \ge 1$ and $P \ge 1$. The optimal solution $\boldsymbol{x}^*$ depends on the relative values of the constraints, yielding three cases:

(1) if $P \le \frac{m}{n}$, then $x_1^* = nk$, $x_2^* = \frac{mk}{P}$, $x_3^* = \frac{mn}{P}$;

(2) if $\frac{m}{n} \le P \le \frac{mn}{k^2}$, then $x_1^* = x_2^* = \left(\frac{mnk^2}{P}\right)^{1/2}$, $x_3^* = \frac{mn}{P}$;

(3) if $\frac{mn}{k^2} \le P$, then $x_1^* = x_2^* = x_3^* = \left(\frac{mnk}{P}\right)^{\frac{2}{3}}$.

This can be visualized as follows:



Proof. By Lemma 3, we can establish the optimality of the solution for each case by verifying that there exist dual variables such that the KKT conditions specified in Def. 3 are satisfied. This optimization problem fits the assumptions of Lemma 3 because the objective function and all but the first constraint are affine functions, which are convex and quasiconvex, and the first constraint is quasiconvex on the positive octant (which contains the intersection of the affine constraints) by Lemma 2.

To match standard notation (and that of Lemma 3), we let $f(\boldsymbol{x}) = x_1 + x_2 + x_3$ and

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} (mnk/P)^2 - x_1x_2x_3 \\ nk/P - x_1 \\ mk/P - x_2 \\ mn/P - x_3 \end{bmatrix}.$$

Thus the gradient of the objective function is $\nabla f(\boldsymbol{x}) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ and the Jacobian of the constraint function is

$$\boldsymbol{J_g}(\boldsymbol{x}) = \begin{bmatrix} -x_2x_3 & -x_1x_3 & -x_1x_2 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Case 1 ($P \le \frac{n}{m}$). We let

$$\boldsymbol{x}^* = \begin{bmatrix} nk & \frac{mk}{P} & \frac{mn}{P} \end{bmatrix}$$

and

$$\boldsymbol{\mu}^* = \begin{bmatrix} \frac{P^2}{m^2nk} & 0 & 1 - \frac{Pn}{m} & 1 - \frac{Pk}{m} \end{bmatrix}$$

and verify the KKT conditions. Primal feasibility is immediate, and dual feasibility follows from $P \le \frac{m}{n} \le \frac{m}{k}$, the condition of this case and by the assumption $n \ge k$. Stationarity follows from direct verification that $\boldsymbol{\mu}^* \cdot \boldsymbol{J_g}(\boldsymbol{x}^*) = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}$. Complementary slackness is satisfied because the only nonzero dual variables are μ_1^*, μ_3^*, and μ_4^*, and the 1st, 3rd, and 4th constraints are tight.

Case 2 ($\frac{m}{n} \le P \le \frac{mn}{k^2}$). We let

$$\boldsymbol{x}^* = \begin{bmatrix} \left(\frac{mnk^2}{P}\right)^{1/2} & \left(\frac{mnk^2}{P}\right)^{1/2} & \frac{mn}{P} \end{bmatrix}$$

and

$$\boldsymbol{\mu}^* = \begin{bmatrix} \left(\frac{P}{mnk^{2/3}}\right)^{3/2} & 0 & 0 & 1 - \left(\frac{Pk^2}{mn}\right)^{1/2} \end{bmatrix}$$

and verify the KKT conditions. The primal feasibility of $x_1 = x_2$ is satisfied because $\frac{nk}{P} \le \frac{mk}{P} \le \left(\frac{mnk^2}{P}\right)^{1/2}$ where the first inequality follows from the assumption $m \ge n$ and the second inequality follows from $m/n \le P$ (one condition of this case). The other constraints are clearly satisfied. Dual feasibility requires that $1 - (Pk^2/mn)^{1/2} \ge 0$, which is satisfied because $P \le mn/k^2$ (the other condition of this case). Stationarity can be directly verified. Complementary slackness is satisfied because the 1st and 4th constraints are both tight for $\boldsymbol{x}^*$, corresponding to the only nonzeros in $\boldsymbol{\mu}^*$.

Case 3 ($\frac{mn}{k^2} \le P$). We let

$$\boldsymbol{x}^* = \begin{bmatrix} \left(\frac{mnk}{P}\right)^{2/3} & \left(\frac{mnk}{P}\right)^{2/3} & \left(\frac{mnk}{P}\right)^{2/3} \end{bmatrix}$$

and

$$\boldsymbol{\mu}^* = \begin{bmatrix} \left(\frac{P}{mnk}\right)^{4/3} & 0 & 0 & 0 \end{bmatrix}$$

and verify the KKT conditions. We first consider the primal feasibility conditions. We have $\frac{nk}{P} \le \frac{mk}{P} \le \frac{mn}{P} \le \left(\frac{mnk}{P}\right)^{2/3}$, where the first two inequalities are implied by the assumption $m \ge n \ge k$ and the last follows from $\frac{mn}{k^2} \le P$, the condition of this case. Dual feasibility is immediate, and stationarity is directly verified. Complementary slackness is satisfied because the 1st constraint is tight for $\boldsymbol{x}^*$ and μ_1^* is the only nonzero.

Note that the optimal solutions coincide at boundary points between cases so that the values are continuous as P varies. □

3.3 Communication Lower Bound

We now state our main result, memory-independent communication lower bounds for general (classical) matrix multiplication with tight constants. After the general result, we also present a corollary for square matrix multiplication.

Theorem 1. *Consider a classical matrix multiplication computation involving matrices of size $n_1 \times n_2$ and $n_2 \times n_3$. Let $m = \max\{n_1, n_2, n_3\}$, $n = \text{median}\{n_1, n_2, n_3\}$, and $k = \min\{n_1, n_2, n_3\}$, so that $m \geq n \geq k$. Any parallel algorithm using P processors that starts with one copy of the two input matrices and ends with one copy of the output matrix and load balances either the computation or the data must communicate at least $D - \frac{mn+mk+nk}{P}$ words of data, where*

$$D = \begin{cases} \frac{mn+mk}{P} + nk & \text{if} \quad 1 \leq P \leq \frac{m}{n} \\ 2\left(\frac{mnk^2}{P}\right)^{1/2} + \frac{mn}{P} & \text{if} \quad \frac{m}{n} \leq P \leq \frac{mn}{k^2} \\ 3\left(\frac{mnk}{P}\right)^{2/3} & \text{if} \quad \frac{mn}{k^2} \leq P. \end{cases}$$

Proof. To establish the lower bound, we focus on a single processor. If the algorithm load balances the computation, then every processor performs mnk/P scalar multiplications, and there exists some processor whose input data at the start of the algorithm plus output data at the end of the algorithm must be at most $(mn + mk + nk)/P$ words of data (otherwise the algorithm would either start with more than one copy of the input matrices or end with more than one copy of the output matrix). If the algorithm load balances the data, then every processor starts and end with a total of $(mn+mk+nk)/P$ words, and some processor must perform at least mnk/P scalar multiplications (otherwise fewer than mnk multiplications are performed). In either case, there exists a processor that performs at least mnk/P multiplications and has access to at most $(mn + mk + nk)/P$ data.

Let F be the set of multiplications assigned to this processor, so that $|F| \geq mnk/P$. Each element of F can be indexed by three indices (i_1, i_2, i_3) and corresponds to the multiplication of $\mathbf{A}(i_1, i_2)$ with $\mathbf{B}(i_2, i_3)$, which contributes to the result $\mathbf{C}(i_1, i_3)$. Let $\phi_{\mathbf{A}}(F)$ be the projection of the set F onto the matrix $\mathbf{A}$, so that $\phi_{\mathbf{A}}(F)$ are the entries of $\mathbf{A}$ required for the processor to perform the scalar multiplications in F. Here, elements of $\phi_{\mathbf{A}}(F)$ can be indexed by two indices: $\phi_{\mathbf{A}}(F) = \{(i_1, i_2) : \exists\ i_3 \text{ s.t. } (i_1, i_2, i_3) \in F\}$. Define $\phi_{\mathbf{B}}(F)$ and $\phi_{\mathbf{C}}(F)$ similarly. The processor must access all of the elements in $\phi_{\mathbf{A}}(F)$, $\phi_{\mathbf{B}}(F)$, and $\phi_{\mathbf{C}}(F)$ in order to perform all the scalar multiplications in F. Because the processor starts and ends with at most $(mn+mk+nk)/P$ data, the communication performed by the processor is at least $|\phi_{\mathbf{A}}(F)|+|\phi_{\mathbf{B}}(F)|+|\phi_{\mathbf{C}}(F)| - \frac{mn+mk+nk}{P}$, which is a lower bound on the communication of the algorithm.

In order to lower bound $|\phi_{\mathbf{A}}(F)| + |\phi_{\mathbf{B}}(F)| + |\phi_{\mathbf{C}}(F)|$, we form a constrained minimization problem with this expression as the objective function and constraints derived from Lemmas 1 and 4. The Loomis-Whitney inequality (Lemma 1) implies that

$$|\phi_{\mathbf{A}}(F)| \cdot |\phi_{\mathbf{B}}(F)| \cdot |\phi_{\mathbf{C}}(F)| \geq |F| \geq \frac{n_1 n_2 n_3}{P} = \frac{mnk}{P},$$

and the lower bound on the projections from Lemma 4 means

$$|\phi_{\mathbf{A}}(F)| \geq \frac{n_1 n_2}{P}, \quad |\phi_{\mathbf{B}}(F)| \geq \frac{n_2 n_3}{P}, \quad |\phi_{\mathbf{C}}(F)| \geq \frac{n_1 n_3}{P}.$$

For any algorithm, the processor's projections must satisfy these constraints, so the sum of their sizes must be at least the minimum value of optimization problem. Then by Lemma 5 (and assigning the projections to x_1, x_2, x_3 appropriately based on the relative sizes of n_1, n_2, n_3), the result follows. □

Corollary 2. *Consider a classical matrix multiplication computation involving two matrices of size $n \times n$. Any parallel algorithm using P processors that starts with one copy of the input data and ends with one copy of the output data and load balances either the computation or the data must communicate at least $3\frac{n^2}{P^{2/3}} - 3\frac{n^2}{P}$ words of data.*

4 CONCLUSION

Theorem 1 establishes memory-independent communication lower bounds for parallel matrix multiplication. By casting the lower bound on accessed data as the solution to a constrained optimization problem, we are able to obtain a result with explicit constants spanning over three scenarios that depend on the relative sizes of the matrix aspect ratios and the number of processors. These constants established in Theorem 1 are tight, as a general 3D algorithm attains the bounds in each of the three scenarios [2]. Our lower bound proof technique tightens the constants proved in earlier work, and we believe it can be generalized to improve known communication lower bounds for other computations.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. CCF-1942892 and OAC-2106920. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 810367).

REFERENCES

[1] A. Aggarwal, A. K. Chandra, and M. Snir. 1990. Communication complexity of PRAMs. *Theor. Comp. Sci.* 71, 1 (1990), 3–28. https://doi.org/10.1016/0304-3975(90)90188-N

[2] H. Al Daas, G. Ballard, L. Grigori, S. Kumar, and K. Rouse. 2022. *Tight Memory-Independent Parallel Matrix Multiplication Communication Lower Bounds.* Technical Report. arXiv. https://doi.org/10.48550/arXiv.2205.13407

[3] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. 2012. Graph expansion and communication costs of fast matrix multiplication. *J. ACM* 59, 6, Article 32 (2012), 23 pages. https://doi.org/10.1145/2395116.2395121

[4] G. Ballard and K. Rouse. 2020. General Memory-Independent Lower Bound for MTTKRP. In *SIAM PP*. 1–11. https://doi.org/10.1137/1.9781611976137.1

[5] S. Boyd and L. Vandenberghe. 2004. *Convex Optimization.* Cambridge University Press. https://web.stanford.edu/~boyd/cvxbook/

[6] J. Demmel, D. Eliahu, A. Fox, S. Kamil, B. Lipshitz, O. Schwartz, and O. Spillinger. 2013. Communication-Optimal Parallel Recursive Rectangular Matrix Multiplication. In *IPDPS*. 261–272. https://doi.org/10.1109/IPDPS.2013.80

[7] J. W. Hong and H. T. Kung. 1981. I/O complexity: The red-blue pebble game. In *STOC*. ACM, 326–333. https://doi.org/10.1145/800076.802486

[8] D. Irony, S. Toledo, and A. Tiskin. 2004. Communication lower bounds for distributed-memory matrix multiplication. *J. Par. and Dist. Comp.* 64, 9 (2004), 1017–1026. https://doi.org/10.1016/j.jpdc.2004.03.021

[9] L. H. Loomis and H. Whitney. 1949. An inequality related to the isoperimetric inequality. *Bull. Amer. Math. Soc.* 55, 10 (1949), 961 – 962. https://doi.org/10.1090/S0002-9904-1949-09320-5

[10] T. M. Smith, B. Lowery, J. Langou, and R. A. van de Geijn. 2019. *A Tight I/O Lower Bound for Matrix Multiplication.* Technical Report. arXiv. https://doi.org/10.48550/arXiv.1702.02017

[11] R. Thakur, R. Rabenseifner, and W. Gropp. 2005. Optimization of Collective Communication Operations in MPICH. *Intl. J. High Perf. Comp. App.* 19, 1 (2005), 49–66. https://doi.org/10.1177/1094342005051521