Acknowledgement

This research is supported in part by NSF grants
ACI-1339676, and DOE grants DOE CS-SC0010200,
DOE DE-SC008700, DOE AC02-05CH11231, ASPIRE Lab industrial
sponsors and affiliates Intel, Google, Hewlett-Packard,
Huawei, LGE, NVIDIA, Oracle, and Samsung.
Other industrial supporters are Mathworks and Cray.
Any opinions, findings, conclusions, or recommendations
in this paper are solely those of the authors
and does not necessarily reflect the position or the
policy of the sponsors.

Parallelepipeds obtaining HBL lower bounds

James Demmel[*] Alex Rusciano[†]

November 13, 2016

This work studies the application of the discrete Hölder-Brascamp-Lieb (HBL) inequalities to the design of communication optimal algorithms. In particular, it describes optimal tiling (blocking) strategies for nested loops that lack data dependencies and exhibit linear memory access patterns. We attain known lower bounds for communication costs by unraveling the relationship between the HBL linear program, its dual, and tile selection. The methods used are constructive and algorithmic. The case when all arrays have one index is explored in depth, as a useful example in which a particularly efficient tiling can be determined.

1. Background: Hölder-Brascamp-Lieb (HBL) Inequalities

HBL inequalities are very powerful and include many famous inequalities, including Hölder's inequality and Young's inequality. Stated for abelian groups $G = \mathbb{Z}^d, G_i = \mathbb{Z}^{d_i}$ with linear maps $\phi_i : G \to G_i$, they take the general form

$$\sum_{x \in G} \prod_{i \in J} f_i(\phi_i(x)) \leq \prod_{j \in J} \|f_i\|_{1/s_i} \tag{1.1}$$

holding for non-negative integrable f_i on G_i. The norms are L_p norms. The s_i for which this holds depend on the maps ϕ_i only, and G for us will be $\mathbb{Z}^d$. We will call such s feasible for the inequality (1.1). It turns out the set of feasible s form a polyhedron we

[*]EECS, Mathematics, University of California, Berkeley, CA (demmel@eecs.berkeley.edu)
[†]Mathematics, University of California, Berkeley, CA (rusciano@math.berkeley.edu)

will denote by $\mathcal{P}$. The case when G or the G_i have a torsion component will not be discussed, although it is of potential interest.

As examples, the commonly stated version of Hölder's amounts to the inequality holding for $0 \leq s_1, s_2 \leq 1$ with $s_1 + s_2 = 1$ and the maps ϕ being the identity maps. Young's inequality for convolutions uses maps from $\mathbb{Z}^2$ to $\mathbb{Z}$. Also it should be noted that HBL inequalities were first introduced and proved for real or complex vector spaces, not abelian groups or rational vector spaces.

To connect this to communication avoidance, the group G is the lattice $\mathbb{Z}^d$ of computations to perform. To perform the computation corresponding to lattice point x, one needs to hold the data contained in $\phi_i(x)$ for all i. The goal of HBL inequalities, in this context, is to bound how many lattice points x we can compute using a memory size of M. This means the f_i are to be the indicator functions of some sets S_i. Then HBL inequalities bound the size of set $S := \cap \phi_i^{-1}(S_i) \subset G$ that are computable provided we store the memory contained in each S_i.

For any feasible s, and memory sizes $M_i = c_i \cdot M$ with $\sum c_i = 1$,

$$|S| \leq \prod |S_i|^{s_i} = \prod_i M_i^{s_i} = M^{1^T s} \cdot \prod c_i^{s_i} \tag{1.2}$$

Inequality (1.2) is a guaranteed upper bound given by the HBL inequality. As an approximation, minimizing the bound amounts to finding the smallest $1^T s$ over all feasible s and neglecting the nuances behind the c_i. We call this minimal sum s_{HBL}. To formally state a rationale for ignoring the $\prod c_i^{s_i}$,

Proposition 1. *Assume M total memory to work with as above and there are n maps. Then for the optimal choice of c_i, the HBL bound 1.2 is $\Theta(M^{1^T s})$. Moreover, the generic choice $c_i = 1/n$ attains this bound.*

Here and in the remainder of the paper, all big O notation is with respect to the memory parameter M, not the dimension of the computation lattice or number of maps.

Proof. Recall the bound in inequality (1.2) is $M^{1^T s} \cdot \prod c_i^{s_i}$. For an upper bound, take the $c_i \leq 1$; no c can do better than this. Now consider $c_i = 1/n$; perhaps there are better c_i for the particular s_i, but we can at least do this well. Consequently, the HBL bound for any s is always in the range

$$[\frac{1}{n^{1^T s}} M^{1^T s}, M^{1^T s}]$$

The lower and upper ranges of this interval are $\Theta(M^{1^T s})$. □

Correspondingly, we will distinguish between two senses of optimality for obtaining the lower bounds.

Definition 2. *The family of sets $S(M)$, parametrized by integer M, form asymptotically optimal tilings if translations of the set $S(M)$ can tile $\mathbb{Z}^d$ and $S(M)$ satisfies*

$$|S(M)| = \Theta(M^{s_{HBL}}) \tag{1.3}$$

as well as

$$\forall i,\ |\phi_i(S(M))| = O(M) \tag{1.4}$$

For exact optimality, while $1^T s$ determines the asymptotic behavior of inequality (1.2), the inequality differs by a constant factor for every c one chooses. Choice of c could be regarded as strategic use of memory. This leads us to consider the sharpness of the following inequality when defining exact optimality:

$$|S| \leq M^{s_{\text{HBL}}} \min_{\substack{s\in\mathcal{P} \\ 1^T s = s_{\text{HBL}}}} \max_{\sum c_i = 1} \prod_i c_i^{s_i}$$

Simple calculations give the optimal choice of c_i. It is equivalent to maximize

$$\sum_i s_i \log(c_i)$$

instead. The method of Lagrange multipliers implies the objective gradient and constraint gradients are parallel:

$$(1,\ldots,1) = \lambda(\frac{s_1}{c_1},\ldots,\frac{s_n}{c_n})$$

The solution to this is $c_i = \frac{s_i}{1^T s}$. This leads to a definition of exact optimality:

Definition 3. *Define scaling parameter*

$$\gamma := \frac{1}{(s_{HBL})^{s_{HBL}}} \min_{\substack{s\in\mathcal{P} \\ 1^T s = s_{HBL}}} \prod_i s_i^{s_i}$$

The family of sets $S(M)$, parametrized by integer M, are exactly optimal tilings if translations of $S(M)$ can tile $\mathbb{Z}^d$ and $S(M)$ satisfies

$$|S(M)| = (1 - o(1)) \cdot \gamma M^{s_{HBL}} \tag{1.5}$$

as well as

$$\forall i,\ \sum_i |\phi_i(S(M))| \leq M \tag{1.6}$$

The $o(1)$ term is required for any reasonable goal because one cannot allocate room for fractions of entries from arrays. Conceptually, we are requiring the ratio of $|S(M)|$ and the theoretical optimum to tend to one, i.e. the relative difference is going to 0.

As another comment on the definition, the minimization problem of computing γ is not difficult; elementary calculus shows that $\log(\prod s_i^{s_i})$ is convex. Indeed, this is the negative entropy function. Minimization of this function can even be done efficiently.

Most of this work focuses on asymptotic optimality, but we will discuss exact optimality in two special cases.

2. Background: Solving for s_{HBL} through an LP

For the maps ϕ, there is a natural necessary condition for the s:

Proposition 4. *If s_i are to satisfy inequality (1.1), then it is necessary that for any subgroup H of G we have the following*

$$\sum s_i \cdot \mathit{rank}(\phi_i(H)) \geq \mathit{rank}(H)$$

Proof. To prove this fact, it suffices to use indicator functions; consequently we show the necessity in the case of Inequality (1.2).

A uniformly growing cube $S(r)$ in the subgroup H parametrized by side length r grows like $r^{\mathrm{rank}(H)}$ in volume because it is a rank(H) dimensional object. This is the LHS of Inequality (1.2).

The RHS of Inequality (1.2) needs to match this growth rate. The images $\phi_i(S(r))$ grow asymptotically like $r^{\mathrm{rank}(\phi_i(H))}$. The LHS must be less than the RHS as $r \to \infty$, implying the result.

The above is more of a sketch; in any case, the above appears in [CDKSY15] as part of Theorem 1.4. □

The surprising part of Theorem 1.4 is that this is actually sufficient. [CDKSY15], supplemented by work from [BCCT], established

Theorem 5. *Given maps ϕ, a collection $s_i \geq 0$ satisfies inequality (1.1) if and only if they satisfy for all subgroups H of G,*

$$\sum_i s_i \cdot \mathit{rank}(\phi_i(H)) \geq \mathit{rank}(H)$$

Because there are only finitely many possible values of the rank of H and its images, there exists a finite list of subgroups which is sufficient to generate the constraints. Consequently the problem can be formulated as a linear program (LP), if we can find a sufficient list of subgroups.

Definition 6 (HBL Primal LP). *If $\mathbf{E}$ is a finite sufficient list of subgroups needed to give the correct HBL constant, we define the HBL primal LP to be*

$$\begin{array}{ll} \textit{minimize} & 1^T s \\ \textit{subject to} & \forall H \in \mathbf{E}, \sum s_i \cdot \mathit{rank}(\phi_i(H)) \geq \mathit{rank}(H) \\ & s_i \geq 0 \end{array}$$

Recent work has started to get a grip on formulating the LP in a computationally feasible manner. In [CDKSY15] a few things are established. For one thing, only the lattice of subgroups generated by $\ker(\phi_i)$ under sums and intersections needs to be used to generate inequality constraints in Theorem 5. However, this lattice is often infinite in higher dimensions. Also, [CDKSY15] describes a terminating algorithm which discovers all the constraints needed to formulate an equivalent LP. However, the algorithm's complexity is unknown. The results of [GGOW] provide a number of novel insights into algorithmic computation of $\mathcal{P}$ arising from the closely related continuous version of the inequalities, including a polynomial time membership and weak separation oracles. Much remains to be understood, and in general formulating and optimizing over $\mathcal{P}$ remains intractable. Here is a summary of tractable special cases, as far as the authors are aware:

- All maps are coordinate projections [CDKSY13].
- Each $\ker(\phi_i)$ is rank 1,2, d-1, or d-2, mixes are allowed [V]. Stated for continuous version of inequalities.
- There are no more than 3 maps; then the kernel subgroup lattice is bound by 28, a classical result [D].

3. Attaining the lower bound by duality

For applications, it is just as interesting to attain lower bounds as to show they exist. In this and the subsequent section, we show how the dual of the HBL Primal LP leads to an asymptotically optimal parallelpiped tiling of the computation lattice.

Notate by $\mathbf{E} = (E_1, \dots, E_k)$ a finite list of subgroups used to formulate the HBL Primal LP. We introduce the notation

$$\operatorname{rank}(\mathbf{E}) := (\operatorname{rank}(E_1), \dots, \operatorname{rank}(E_k))^T,$$

and similarly

$$\operatorname{rank}(\phi_i(\mathbf{E})) := (\operatorname{rank}(\phi_i(E_1)), \dots, \operatorname{rank}(\phi_i(E_k)))^T$$

One may write the dual of the HBL Primal LP as

$$\begin{array}{cc} \text{maximize} & y^T \operatorname{rank}(\mathbf{E}) \\ \text{subject to} & \forall \phi_i,\ y^T \operatorname{rank}(\phi_i(\mathbf{E})) \leq 1 \\ & y_i \geq 0 \end{array}$$

It will be useful to be more flexible in how we think of the dual problem. The dual, as formulated from the primal, comes with a particular subgroup list $\mathbf{E}$ indexing the dual variables. The methods we use add and remove subgroups from consideration, and we do not wish this to fundamentally change the dual LP. In the future we use the notation $\mathbf{E}$ for the analogous, but more generic, role in the revised dual:

Definition 7 (Dual LP). *Recall the HBL setting consists of maps ϕ_i from lattice $\mathbb{Z}^d$. A dual vector y will be considered to be indexed by all subgroups of $\mathbb{Z}^d$, but with finitely* *many non-zero coordinates. The non-zero coordinates are defined to be the support of y. If the list of subgroups $\mathbf{E} = (E_1, \ldots, E_t)$ is the support of y, then we introduce a few notations and definitions. Two natural notational shorthands are*

$$y^T rank(\mathbf{E}) := \sum y_{E_j} rank(E_j)$$

and

$$y^T rank(\phi_i(\mathbf{E})) := \sum y_{E_j} rank(\phi_i(E_j))$$

Now define the objective value of y to be

$$val(y) := y^T rank(\mathbf{E}) \tag{3.1}$$

and say y is feasible if it satisfies the conditions

$$\forall \phi_i,\ C_i(y) := y^T rank(\phi_i(\mathbf{E})) \leq 1 \tag{3.2}$$

$$y_i \geq 0 \tag{3.3}$$

As a further notational note on this definition, we use a few symbols in place of $\mathbf{E}$ when extra information is present. Typically we will use $\mathbf{Y}$ when the supporting subgroups are independent, and $\mathbf{U}$ when they are a flag. These definitions are covered later.

This is readily interpretable when the supporting subgroups $\mathbf{Y}$ of the dual vector are independent. Here independent means that $\text{rank}(\oplus_i Y_i) = \sum_i \text{rank}(Y_i)$. Before explaining exactly how we interpret the dual, we need the following geometrically intuitive lemma. It demonstrates that asymptotic optimality eases some difficulties stemming from discreteness.

Lemma 8. *Take any independent elements $e_1, \ldots, e_h$ contained in rank h subgroup $Y \subset \mathbb{Z}^d$ and linear mapping L. Define the set*

$$S := \{z \in \mathbb{Z}^d | z = \sum a_i e_i \text{ with } 0 \leq a_i \leq \lfloor M^k \rfloor - 1,\ a_i \in \mathbb{Z}\} \tag{3.4}$$

In this equation, k is an arbitrary positive number, and M is an integer conceptually representing memory capacity. In applications later, $k \in (0, 1]$.

Then for any linear map L, $|S| = \Theta(M^{kh})$ and $|L(S)| = O(M^{kr})$ where $r := rank(L(Y))$. In applications later, L is taken to be one of the ϕ_i.

Proof. The elements in set S are $O(M^k)$ from the origin in Euclidean distance, hiding the dimensional factor d in the big O notation. By linearity, the elements of $L(S)$ are also $O(M^k)$ from the origin in $\text{im}(L)$. Therefore an r dimensional cube residing within $L(Y)$ with side lengths $O(M^k)$ can contain $L(S)$. This means that $|L(S)| = O(M^{kr})$.

Finally, from independence of the e_i it follows that $|S| = \lfloor M^k \rfloor^h$ □

We now define the parallelepiped-like construction that will be used to create asymptotically optimal tilings. Although not the only possible way to build good tiling shapes, it is flexible and leads to clean descriptions.

Definition 9 (Product Parallelepiped)**.** *Suppose we are given a dual vector y, whose non-zero values are attached to a set of independent subgroups $Y_1, \dots, Y_t$. Form S_{Y_i} as in Eq. 3.4, using $k = y_{Y_i}$.*

Now define a parallelepiped shaped tile through a Minkowski sum of sets

$$S := S_{Y_1} + \cdots + S_{Y_t} \tag{3.5}$$

The independent elements used to construct S_{Y_i} are left unspecified; the choice affects constants, but will not affect asymptotic optimality.

To make things explicit, Algorithm 1 below produces the translations needed to tile $\mathbb{Z}^d$ with set S. It uses an important matrix factorization for linear maps between abelian groups (or more generally between modules over a principle ideal domain) known as the Smith Normal Form.

The Smith Normal Form of a matrix A with integer entries is of the form $A = UDV^{-1}$ where U, V are unimodular and D is diagonal with non-negative integer entries. Its diagonal entries $d_i := D_{ii}$ are uniquely defined by requiring $d_i | d_{i+1}$. In our use of the factorization, the matrix A is injective and consequently full rank, so that all d_i are non-zero.

Algorithm 1 Construct tile S and its translations T that tile $\mathbb{Z}^d$

```
 1: Input: Memory parameter $M$
 2: Input: For each $i = 1, \dots, t$: independent elements $e_{i1}, \dots, e_{ih_i}$ chosen from independent rank $h_i$ subgroups $Y_i$
 3: Input: For each $i = 1, \dots, t$: memory scaling parameter $k = y_{Y_i}$ for subgroup $Y_i$
 4: Output: translations $T$ of the set $S$ that tile $\mathbb{Z}^d$
 5: $E \leftarrow (e_{11}, e_{12}, \dots, e_{th_t})$
 6: $S \leftarrow \{E \cdot (a_{11}, a_{12}, \dots, a_{th_t})^T | \, a_{ij} \in \{0, \dots, \lfloor M^{y_{Y_i}} \rfloor - 1\}$
 7: $m \leftarrow \sum h_i$
 8: $(U, D, V) \leftarrow$ Smith Normal Form$(E)$
 9: $U' \leftarrow$ last d-m columns of U
10: $U'' \leftarrow$ first m columns of U
11: $T_1 \leftarrow \{E \cdot (a_{11}, a_{12}, \dots, a_{th_t})^T \,|\, a_{ij} \in \lfloor M^{y_{Y_i}} \rfloor \cdot \mathbb{Z}\}$
12: $T_2 \leftarrow \{U' \cdot (a_1, \dots, a_{d-m})^T \,|\, a_i \in \mathbb{Z}\}$
13: $T_3 \leftarrow \{U'' \cdot (b_1, \dots, b_m)^T \,|\, b_i \in \{0, \dots, d_i - 1\}\}$
14: $T \leftarrow$ Minkowski sum $T_1 + T_2 + T_3$
        return $S, T$
```

The set S returned by the algorithm exactly follows Def. 9. The translations T come from two sources: T_1 accounts for the finite size of M while tiling the subgroup generated by $e_{11}, e_{12}, \dots, e_{th_t}$ under integer linear combinations. In the future we will write this subgroup as $\langle e_{11}, e_{12}, \dots e_{th_t} \rangle$, and similarly for other generated subgroups. The others T_2, T_3 account for the need to tile each coset in $\mathbb{Z}^d / \langle e_{11}, e_{12}, \dots e_{th_t} \rangle$.

Proposition 10. *Algorithm 1 correctly outputs a parallelepiped set S which under translation by T tiles $\mathbb{Z}^d$. This holds for any input: that is, for any selection of independent subgroups U_i, choice of independent elements within these subgroups, memory parameter setting M, and memory scalings $M^{y_{Y_i}}$.*

Proof. Let us begin by discussing the translations in T_1. Consider $x = \sum a_{ij} e_{ij}$. The set S only uses scalings of 0 to $\lfloor M^{y_{Y_i}} \rfloor - 1$ of each e_{ij}. Consequently, for x to be contained in a translation of S, the shift component in the e_{ij} direction must be in the range

$$[a_{ij} - \lfloor M^{y_{Y_i}} \rfloor + 1, a_{ij}]$$

The only such member of T_1 has e_{ij} component $\lfloor a_{ij} / \lfloor M^{y_{Y_i}} \rfloor \rfloor \cdot \lfloor M^{y_{Y_i}} \rfloor$

T needs to also contain exactly one representative of each coset in

$$\mathbb{Z} / \langle e_{11}, e_{12}, \dots e_{th_t} \rangle \simeq \mathbb{Z}^{d-m} \oplus (\bigoplus_{i=d-m}^{d} \mathbb{Z}/D_i\mathbb{Z})$$

T_2 accounts for the free component, and T_3 for the torsion component.

Indeed, let $U'a + U''b, U'a' + U''b'$ be two distinct elements of $T_2 + T_3$. Saying they are in the same coset is exactly saying their difference lies in $\mathrm{im}(E)$. Writing $E = UD(V)^{-1}$ as in Algorithm 1 and noting V is unimodular, it is clear that $\mathrm{im}(E) = \mathrm{im}(UD)$. Conclude that lying in the same coset is equivalent to

$$U'a + U''b - U'a' - U''b' \in \mathrm{im}(UD)$$

As U is unimodular, This means for some $c \in \mathbb{Z}^d$

$$(b, a)^T - (b', a')^T = Dc$$

Because D is d-by-m, the the last $d - m$ coordinates of Dc are 0. This means $a = a'$. Also for b, b' to be used in T_3, they must satisfy $0 \le b_i, b'_i < d_i$. But then

$$-d_i < b_i - b'_i = d_i c_i < d_i$$

which is only possible if $b_i = b'_i$.

To conclude that all cosets are represented, we argue that for any $x \in \mathbb{Z}^d$, there are $U'a, U''b$ such that $x - U(b,a)^T \in \mathrm{im}(E)$. Take $b_i = (U^{-1}x)_i \bmod (D_i)$ and $a_i = (U^{-1}x)_{m+i}$.

□

This paper includes examples at the end in Appendix A. These will help demonstrate this approach and future aspects of the paper.

Now that the fundamental tiling object and mechanism have been described, we now begin to analyze the properties of the tile S in relation to the HBL problem.

Proposition 11. *Suppose we are given a dual vector y, whose non-zero values are attached to a list of independent subgroups $\mathbf{Y} = (Y_1, \ldots, Y_t)$. Form the product parallelepiped S of Def. 9.*

Then $|S| = \Theta(M^{y^T rank(\mathbf{Y})})$. If in addition y is dual feasible, then $|\phi_i(S)| = O(M)$ holds for each ϕ_j.

Proof. By independence of the subgroups contained in $\mathbf{Y}$, it follows that $|S| = \prod_i |S_{Y_i}|$. Apply the count estimates of Lemma 8 to this:

$$|S| = \prod_i \Theta(M^{\text{rank}(Y_i) \cdot y_{Y_i}}) = \Theta(M^{y^T \text{rank}(\mathbf{Y})})$$

It remains to consider the images of this set under the ϕ_j in the case y is feasible. This requires a bound on $|\phi_i(S)|$ Invoking the count estimates of Lemma 8 in the second inequality, and feasibility property (3.2) of y in the third inequality

$$|\phi_j(S)| \leq \prod_i |\phi_j(S_{Y_i})| \leq \prod_i M^{\text{rank}(\phi_j(Y_i)) y_{Y_i}} = M^{C_j(y)} \leq M$$

□

It will be necessary to strengthen the bounds on the $|\phi_i(S)|$ later, but this already proves a useful result:

Corollary 12. *Suppose there exists a dual optimal solution y with non-zero dual variables attached to a set of independent subgroups $\mathbf{Y}$. Then we may tile the lattice $\mathbb{Z}^d$ with an asymptotically optimal parallelpiped shape.*

Proof. From strong duality, $y^T \text{rank}(\mathbf{Y}) = s_{HBL}$. Then Proposition 11 provides a parallelepiped shape that is asymptotically optimal, as it satisfies equations 1.3 and 1.4. Algorithm 1 shows how to tile using object S. □

This construction breaks down when dual variables correspond to non-independent subspaces. The next section generalizes the results of this section through a notion of flags of the subgroup lattice.

4. Attainability in General

In this section, we describe an algorithm for producing an asymptotically optimal tiling. The recipe is to formulate the primal, solve the corresponding dual, and iteratively modify the solution of the dual to something geometrically interpretable. Consequently, at least asymptotically, the HBL lower bounds are attainable by a polyhedral tiling, and to do so is essentially no harder than describing the set of feasible s for inequality (1.1). This set is commonly referred to as the Brascamp-Lieb Polyhedron.

4.1. Flags

It might not be possible to find a dual vector supported on independent subgroups that obtains the optimal value. However, it turns out that it is possible to find one supported on what we here define to be a flag.

Definition 13. *A flag of the lattice $\mathbb{Z}^d$ (for us) is a sequence $\mathbf{U}$ of strictly nested subgroups*

$$\emptyset \subset U_1 \subset \cdots \subset U_t = \mathbb{Z}^d$$

We want to take the dual solution, and transform it to being supported on a flag. The following is a simple but important property in accomplishing this goal. It was also helpful in =[CDKSY15] and [V], the latter of whom we note found flags useful in studying the vertices of the Brascamp-Lieb polyhedron.

Lemma 14 (Substitution Lemma)**.** *For any linear map L on $\mathbb{Z}^d$ and subgroups V, W,*

$$rank(L(V)) \geq rank(L(V \cap W)) + rank(L(V + W)) - rank(L(W))$$

On the other hand,

$$rank(V) = rank(V \cap W) + rank(V + W) - rank(W)$$

Proof. The claimed equality in the lemma follows by writing a basis for $V \cap W$ and completing it to a basis for W with a second set of independent basis elements. Call the subgroup spanned by the second set P. Observe P has trivial intersection with V, and the rank of P is $\text{rank}(W) - \text{rank}(V \cap W)$. Applying these observations to $W + V = P + V$,

$$\text{rank}(W+V) = \text{rank}(P+V) = \text{rank}(P) + \text{rank}(V) = \text{rank}(W) - \text{rank}(W \cap V) + \text{rank}(V)$$

establishing the result. To prove the inequality, apply the equality to subspaces $L(V)$, $L(W)$, and then observe

$$L(V \cap W) \subseteq L(V) \cap L(W), \text{ while } L(V + W) = L(V) + L(W)$$

The reason for the possible inequality is that maybe there are different elements $v \in V$ and $w \in W$, but $L(v) = L(W)$. $\square$

We employ this observation repeatedly to shift the support of a dual vector onto a flag, through the following procedure. It takes as input a feasible y supported on an arbitrary list $\mathbf{E}$ and outputs a feasible y' supported on a flag $\mathbf{U}$ with the same objective value Eq. 3.1. Recall by feasible we mean Eq. 3.2, 3.3 are satisfied.

Algorithm 2 Find dual feasible vector supported on a flag

```
 1: Input: dual feasible vector $y$ supported on $E_1, \dots, E_m$
 2: Output: feasible $y'$ supported on a flag $U_1, \dots U_t$, with the same objective value as $y$
 3: Initialize $y'$ as $y$
 4: while $y'$ is not supported on a flag do
 5:     $V, W \leftarrow$ any pair in the support of $y'$ NOT satisfying $V \subset W$ or $W \subset V$
 6:     Let $V$ be the member of the pair with $y'_V \leq y'_W$
 7:     $y'_W \leftarrow y'_W - y'_V$
 8:     $y'_{V+W} \leftarrow y'_{V+W} + y'_V$
 9:     $y'_{V\cap W} \leftarrow y'_{V\cap W} + y'_V$ (if $V \cap W \neq \{0\}$)
10:     $y'_V \leftarrow 0$
    return $y'$ and its support
```

Theorem 15 (Non-negative Flag Theorem)**.** *Algorithm 2 is correct: given input a dual feasible vector y supported on $E_1, \dots, E_m$, it outputs a dual feasible y' supported on a flag $\mathbf{U} = (U_1, \dots U_t)$ with the same objective value as y.*

Proof. The existence of the pair V, W is equivalent to the support of y' not being totally ordered, which is equivalent to the support of y' not being a flag. So if the algorithm does terminate, the support will be a flag. We must show the algorithm terminates, and that y' maintains the objective value and feasibility.

Induction establishes that y' is always non-negative. Indeed, inside the while loop, the only danger is $y'_W - y'_V$. But y'_V is the smaller of the two by construction. So y' always satisfies Eq. 3.3.

Let y'' denote the value of y' after another pass through the while loop. We examine the effect of the iteration on Eq. 3.2, 3.1. In the case $V \cap W \neq \{0\}$,

$$C_i(y'') = C_i(y') - y'_V\left[\operatorname{rank}(\phi_i(W)) - \operatorname{rank}(\phi_i(V\cap W)) - \operatorname{rank}(\phi_i(V+W)) + \operatorname{rank}(\phi_i(V))\right]$$

The bracketed quantity is non-negative by Lemma 14, meaning Eq. 3.2 still holds. If $V \cap W = \{0\}$, then

$$\begin{aligned} C_i(y'') &= C_i(y') - y'_V\left[\operatorname{rank}(\phi_i(W)) - \operatorname{rank}(\phi_i(V+W)) + \operatorname{rank}(\phi_i(V))\right] \\ &= C_i(y') - y'_V\left[\operatorname{rank}(\phi_i(W)) - \operatorname{rank}(\phi_i(V\cap W)) - \operatorname{rank}(\phi_i(V+W)) + \operatorname{rank}(\phi_i(V))\right] \end{aligned}$$

where we used $\operatorname{rank}(\phi_i(V \cap W)) = 0$. Consequently Lemma 14 applies again. Similarly, the objective value is preserved: in the case of $W \cap V \neq \{0\}$,

$$\operatorname{val}(y'') = \operatorname{val}(y') - y'_V\left[\operatorname{rank}(W) - \operatorname{rank}(V\cap W) - \operatorname{rank}(V+W) + \operatorname{rank}(V)\right]$$

with the bracketed quantity being 0 by Lemma 14. As before, the same follows in the case $V \cap W = \{0\}$ by noting $\text{rank}(V \cap W) = 0$.

It remains to establish that the algorithm will terminate. At first glance, it appears that the y' might cycle in the algorithm. However, each iteration is increasing the dual variables on $V + W$ and $V \cap W$, so the dual vector seems to be shifting towards the high and low rank subgroups.

To capture this intuition, we define a simple measure of extremeness on dual vectors. Recall all groups reside in $\mathbb{Z}^d$. To a dual vector y we assign a list $w(y)$ of length d. To do this, set

$$w(y)_i = \sum_{U \in \text{ support}(y),\ \text{rank}(U)=i} y_U$$

For example, if y is supported on $\langle e_1, e_2 \rangle, \langle e_1 \rangle, \langle e_2 \rangle$ with values $1, .5, 2$, and $d = 3$, then $w(y) = (2.5, 1, 0)$. We say y' is *more extreme* than y'' if $w(y')$ is reverse lexicographically more than $w(y'')$. Every iteration of the while loop makes y' more extreme; indeed, the value y_{V+W} increases and $V + W$ is of strictly larger rank than V or W.

Now we show that $w(y')$ can take on only finitely many values, completing the proof. Observe that $y'^T 1$ stays the same or decreases each iteration, so coordinates of $w(y)$ are bound by $y'^T 1$. Also, the values produced by the algorithm come from performing only addition and subtraction operations on the the coordinates of y, which are rational. Consequently coordinates of $w(y')$ lie in the finite set

$$\text{span}_{\mathbb{Z}}(y_{E_1}, \ldots, y_{E_m}) \cap [0, y^T 1]$$

□

4.2. Parallelepiped Tilings from Flags

The main theorem of the previous section allows us to transform an optimal dual vector into another optimal dual vector supported on a flag. Now we convert the flag subgroups into independent subgroups in the natural manner in order to produce a tiling shape.

Definition 16 (Flag Parallelepiped)**.** *Suppose y is supported on flag $\mathbf{U}$. Let $\mathbf{Y}$ be a sequence of independent subgroups such that $Y_1 + \cdots + Y_i = U_i$. Define the dual vector y' supported on $\mathbf{Y}$ by*

$$y'_{Y_i} = y_{U_i} + \cdots + y_{U_t}$$

Form a product parallelepiped S of Def. 9 from y'. We will call S the flag parallelepiped of y, and y' its associated dual vector.

Here let's briefly summarize the progress so far, and what we still need to accomplish. Provided we formulated the HBL Primal LP and solved its dual, we found a feasible

y with objective value s_{HBL}. From Theorem 15, this y can be modified to another y' supported on some flag, maintaining the objective value s_{HBL} and feasibility as defined in Eq. 3.1, 3.2, 3.3. Next apply the flag parallelepiped construction of Def. 16 to y' to create a tile S and its associated y''. Proposition 11 implies that S includes $\Theta(M^{s_{HBL}})$ lattice points. However, y'' might no longer satisfy Eq. 3.2, so Proposition 11 does not show that $|\phi_i(S)| = O(M)$. We need to expand the analysis of Lemma 8 to the case of parallelepipeds instead of cubes.

Lemma 17 (Growing Parallelepiped Lemma)**.** *Consider independent subspaces $Y_1, \ldots, Y_t$ with corresponding dual values y_{Y_i}. Construct the product parallelepiped as in Def. 9 from these independent spaces and dual values. Assume the subgroups are ordered so that y_{Y_i} monotonically decreases with i. In keeping with Def. 16 have $U_i := Y_1 + \cdots + Y_i$ for $i = 1, \ldots, t$, and for convenience $U_0 := \{0\}$. For any linear map L, set*

$$d_i := \mathit{rank}(L(U_i)) - \mathit{rank}(L(U_{i-1}))$$

Then we have the bound

$$|L(S)| = O\left(\prod M^{y_{Y_i} \cdot d_i}\right)$$

In particular, this holds for L chosen to be any of the ϕ_j.

Before beginning the proof, we remark on the significance. The weaker bound used in Proposition 11 was

$$|L(S)| \leq \prod |L(S_{Y_i})| = O\left(\prod M^{y_{Y_i} \cdot a_i}\right)$$

with $a_i := \text{rank}(L(Y_i))$. From independence of the subgroups Y_j, it is immediate that $d_i \leq a_i$. For example, when L is the identity, $d_i = a_i$. However, when $L(Y_i)$ is not independent of $L(U_{i-1})$, it is always the case that $d_i < a_i$.

Proof. The goal is to propose a rectangular prism T containing $L(S)$. Of the defining edges, d_i of them will be length $O(M^{y_{Y_i}})$. This would prove the needed bound.

Intuitively, we just need to make the d_1 dimensions coming from $L(Y_1)$ have the largest size $O(M^{y_{Y_1}})$, and the next d_2 dimensions coming from Y_2 will need to have length $O(M^{y_{Y_2}})$ and so forth. To formally show this by constructing T, it is convenient to interpret all subgroups instead as subspaces of $\mathbb{Q}^d$ with the standard Euclidean inner product and its induced norm. Now apply a Gram-Schmidt orthogonalization procedure to the sequence $L(Y_1), L(Y_2), \ldots, L(Y_t)$. This yields subspaces $E_1, \ldots E_t$ satisfying

$$E_1 = L(Y_1),\ E_1 + \cdots + E_i = L(Y_1) + \cdots + L(Y_i),\ E_i \perp E_j \,\text{for}\, i \neq j$$

Take T to be the Minkowski sum formed by cubes T_i of side length $O(M^{y_{Y_i}})$ growing in the spaces E_i. It is readily observed that $|T| = O\left(\prod M^{y_{Y_i} \cdot d_i}\right)$. Denote by P_{E_i} the orthogonal projection onto E_i. If we can show $P_{E_i}(L(S)) \subset T_i$ for each i, then $L(S) \subset T$. The proof would then be complete.

Select an arbitrary $x \in S$. That is,

$$L(x) = L(x_{Y_1}) + \cdots + L(x_{Y_t})$$

where $x_{Y_j} \in S_{Y_j}$. Observe that $L(x_{Y_i}) \in \ker(P_{E_j})$ for $i < j$. Also $\|L(x_{Y_j})\|_2 = O(M^{y_{Y_j}})$. This implies

$$P_{E_i}(L(x)) = P_{E_i}(L(x_{Y_i})) + \cdots + P_{E_i}(L(x_{Y_t}))$$

and therefore

$$\|P_{E_i}(L(x))\|_2 = O(M^{y_{Y_i}}) + \cdots + O(M^{y_{Y_t}}) = O(M^{y_{Y_i}})$$

As T is permitted to be $O(M^{y_{Y_i}})$ in $E_i = \mathrm{im}(P_i)$, we conclude that $P_i(S) \subset T$ if the hidden constant for T large enough. $\square$

This readily applies to the construction of Def. 16:

Theorem 18 (Non-negative Parallelepiped Theorem). *From an optimal dual feasible vector y supported on a flag $\mathbf{U}$, form a flag parallelepiped S. Then $|\phi_j(S)| = O(M)$ for each ϕ_j in the HBL problem, and $|S| = \Theta(M^{s_{HBL}})$.*

Proof. Let y' be the associated dual vector of S with independent subgroups $Y_1, \ldots, Y_t$ as described in Def. 16. As y has positive entries, y'_{Y_i} are monotonically decreasing. Consequently, Lemma 17 applies. It implies that

$$\begin{aligned}|\phi_j(S)| = O(\prod_i M^{y'_{Y_i} \cdot d_i}) &= O(\prod_i M^{(y_{Y_i} + \cdots + y_{Y_t}) \cdot d_i}) \\ &= O(M^{\sum_i y_{Y_i} \cdot (d_1 + \cdots + d_i)}) \qquad = O(M^{y^T \mathrm{rank}(\phi_j(U))}) = O(M)\end{aligned}$$

That $|S| = \Theta(M^{s_{\mathrm{HBL}}})$ follows from Proposition 11 $\square$

This is the major theoretical result. Combined with earlier results, it notably establishes the central claim:

Corollary 19. *If one is able to produce a sufficient list of subgroups for the HBL primal, then one can determine an asymptotically optimal tiling shape.*

Proof. Solve the dual LP to get y. Apply Theorem 15 to produce y' optimal and supported on a flag. Then apply the flag parallelepiped construction to y' yielding tile S, which is asymptotically optimal by Theorem 18. $\square$

5. Two Cases of Exactly Optimal Tilings

The preceding sections have focused on attaining asymptotic optimality. We would like the tiling shape to meet the requirements of exact optimality, as given by equations 1.5 and 1.6. This would matter for communication avoiding applications in practice. However, we do not have a characterization of when this is possible. Instead, we will describe two simpler situations in which we can provide exactly optimal tilings.

5.1. Rank One Maps

We begin by discussing the case of rank 1 maps. This could be regarded as a generalized n-body problem. Detailed work on the communication patterns and bounds for the n-body problem was examined in [DGKSY] and [KY]. This corresponds to arrays with single indices. First, we demonstrate what s_{HBL} is for this case and a method for obtaining asymptotic optimality.

Proposition 20. *Assume the maps ϕ_i are rank 1 with $i \in J$ and $|J| = n$, and the lattice is $\mathbb{Z}^d$. If $\cap_i \, ker(\phi_i) = \emptyset$, then $s_{HBL} = d$. Then a d dimensional cube with sides $O(M)$ is asymptotically optimal. Otherwise, the Primal LP of Def. 6 is infeasible.*

Proof. First, suppose that $\cap_i \ker(\phi_i)$ is nonempty. Then take E_1 to be a non-zero element of this intersection; as kernels are subspaces, the $\langle E_1 \rangle$ is also in the kernel. This implies the corresponding constraint in Def. 6 is

$$0^T s \geq 1$$

which can't be satisfied. So the LP is infeasible, meaning one could get "infinite" data re-use. See the appendix for an example.

Now suppose $\cap \ker(\phi_i) = \emptyset$. One subgroup you could use is $\mathbb{Z}^d$ itself. By the rank 1 assumption, the inequality constraint in Def. 6 corresponding to this subgroup is

$$1^T s \geq d$$

This implies $s_{HBL} \geq d$. Now we exhibit a feasible primal vector s for which $1^T s = d$ to complete the proof. One may select a subset $J' \subset J$ with $|J'| = d$ such that $\cap_{i \in J'} \ker(\phi_i) = \emptyset$. This follows by induction; start with $H_0 = \mathbb{Z}^d$. Then recurse by $H_i = H_{i-1} \cap \ker(\phi_i)$. If $\text{rank}(H_i) = \text{rank}(H_{i-1}) - 1$ then include i in J'. Because belonging to $\ker(\phi_i)$ amounts to satisfying a single linear equation, the rank may only decrease by 1. Choose the primal variable s to be $1_{J'}$.

It remains to establish the feasibility of this s. The argument may proceed recursively as above. This time label the elements of J' to be $i_1, \ldots, i_d$, and let T denote any subgroup. Set $H_{i_0} = T$ and $H_{i_j} = H_{i_{j-1}} \cap \ker(\phi_i)$ recursively. Again, ranks of the H_{i_j} decrease by 1 or stay the same, compared to the rank of $H_{i_{j-1}}$. The end result is $\{0\}$; this implies T

is not a strict subset of at least $\text{rank}(T)$ of the kernels associated with J'. Consequently s satisfies the constraint of Def. 6 for the subgroup T:

$$s^T \text{rank}(\phi(T)) \geq \text{rank}(T)$$

This implies the feasibility of s and establishes $s_{HBL} = d$. Observe the dual variable indicating the space $\mathbb{Z}^d$ achieves the value s_{HBL} as well. By Lemma 8, a cube with sides $O(M)$ is asymptotically optimal. □

This establishes that the running Algorithm 1 on $\mathbb{Z}^d$ produces an asymptotically optimal tiling. For exact optimality, we must restrict to the case where there are d rank-one maps with empty kernel intersection.

Lemma 21 (Basis Lemma)**.** *The subgroup $\cap_{j \neq i} ker(\phi_j)$ is rank 1; take e_i to be a non-zero element of smallest Euclidean norm from this subgroup. Then each subgroup $ker(\phi_i)$ contains the independent elements $e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_d$.*

Proof. We must check the e_i are well defined, and that they are linearly independent.

Every time we intersect with one of the kernels, the rank reduces by 1. The empty intersection property of the d kernels implies this; every intersection adds a linear constraint, and if one of the linear constraints turned out to be redundant then d intersections would not result in the empty set.

Lastly, we make sure that the e_i are independent. If not, then some e_i is in the span of the other $e_{i'}$; however, the other $e_{i'}$ are contained in $\ker(\phi_i)$. This means $e_i \in \ker(\phi_i)$ is as well. Then e_i lies in the intersection of all the kernels, which by assumption is empty set. □

This basis is critical in the following;

Proposition 22. *Let e_i be as in Lemma 21. Then the sets $S := \{\sum a_i e_i | a_i \in \mathbb{Z}, 0 \leq a_i \leq \lfloor M/d \rfloor - 1\}$ are exactly optimal. That is, the output of Algorithm 1 on independent elements $e_1, \ldots, e_d$ of $\mathbb{Z}^d$ meets the requirements of Eq. 1.5 and 1.6.*

Proof. The first part of this section established that the optimal s_{HBL} is d and comes from each $s_i = 1$. This is in fact the unique solution to the primal LP of Def. 6 so it is by default the minimizer of γ in Eq. 1.5. Alternatively, evenly distributed values s_i maximize entropy and consequently would minimize γ. Plugging this in, $c_i = \frac{1}{d}$ and $\gamma = \frac{1}{d^d}$.

It remains to confirm that $|S| = (M/d)^d + O(1)$ and $\sum_i |\phi_i(S)| \leq M$. First, by independence of the e_i, there are $\lfloor M/d \rfloor^d$ lattice points enclosed. Now if $M = a \cdot d + r$,

$$(M/d)^d = a^d \cdot (1 + \frac{r}{M})^d = \lfloor M/d \rfloor^d \cdot (1 + \frac{r}{M})^d \leq \lfloor M/d \rfloor^d e^{r/M} = \lfloor M/d \rfloor^d (1 + o(1))$$

This establishes Eq. 1.5. For the memory bound constraint, consider $\phi_i(S)$. Applied to any point $z \in S$, it outputs $a_i \cdot e_i$. As a_i only varies between $\lfloor M/d \rfloor$ values, the result follows.

□

We may summarize the approach to tiling in Proposition 22 in the the following algorithm.

Algorithm 3 Exactly Optimal Tiling, Rank One Maps

```
1: Input: rank one maps $\{\phi_i\}_{i=1}^d$ with coordinate representations $a_i \in \mathbb{Z}^d$, satisfying
   $\cap \ker(\phi_i) = \{0\}$, memory parameter $M$
2: Output: tile $S$ and translations by $T$ that tile $\mathbb{Z}^d$
3: Initialize $e_1, \dots, e_d \in \mathbb{Z}^d$
4: for i = 1 to d do
5:     $A \leftarrow (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_d)^T$
6:     $U$, $D$, $V \leftarrow$ Smith Normal Form($A$)
7:     $e_i \leftarrow$ column 1 of $V$
8: $S$, $T \leftarrow$ Algorithm 1 on input subgroup $U_1 = \mathbb{Z}^d$, its independent elements $e_1, \dots, e_d$,
   memory parameter $M$, and scaling $y_{\mathbb{Z}^d} = 1$
       return $S$ and $T$
```

The new component of the algorithm is calculating the independent elements e_i. With this in mind, we examine the calculation of the e_i. Recall $e_i \in \cap_{j \neq i} \ker(\phi_j)$ of smallest Euclidean norm are used in Proposition 22. The implies e_i is in the kernel of matrix $A_i := (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_d)^T$. Decompose this matrix by Smith Normal Form, giving the representation UDV^{-1}. Because the rank of A_i is $d-1$, only the first diagonal entry of D is 0. This means the kernel of the matrix is exactly what V^{-1} maps to $(1, 0, \dots, 0)^T$, meaning multiples of the first column of V. As V is unimodular, this column is also the shortest integer valued multiple of itself.

5.2. Rank d-1 Maps

This section follows the rank 1 case very closely, and consequently is kept brief. As an example, this setting includes the case of matrix multiplication and therefore much of linear algebra. We again discuss asymptotic optimality, followed by exact optimality.

Proposition 23. *In the case where all maps are rank $d-1$, the optimal dual vector y can be taken to have the subspace generated by the kernels of all the maps as its only nonzero coordinate. Call this subspace H and let $k = rank(H)$. Then $y_W = 1/(k-1)$ is optimal for the dual LP of Def. 7. In addition, $s_{HBL} = k/(k-1)$.*

Proof. As H is sent to a rank $k-1$ space by each of the ϕ_i, y is indeed dual feasible with objective value $k/(k-1)$.

We must show that that this matches the HBL lower bound, as then by strong duality

y is dual optimal. Propose the primal value $s = 1/(k-1) \cdot 1_A$, where 1_A indicates any k maps whose (one-dimensional) kernels generate the rank k space. Essentially, this is saying the kernels of these maps are independent.

Then consider any rank l subgroup T, and its images under the maps in A. By independence of kernels in the construction of A, only l of the maps might send this to a rank $l-1$ group, the others send it to a l dimensional space. As there are k non-zero s_i, the LHS of the constraint given by subgroup T in the primal LP of Def. 6 is

$$\begin{aligned}
\operatorname{rank}(\phi(T))^T s &= \sum_{\phi_i \in A} s_i \cdot \operatorname{rank}(\phi_i(T)) \\
&= \frac{1}{k-1} \sum_{\phi_i \in A} \operatorname{rank}(\phi_i(T)) \\
&= \frac{1}{k-1} \left[(l-1) \cdot \#\{\phi \in A | \ker(\phi) \cap T \neq \{0\}\} + l \cdot \#\{\phi \in A | \ker(\phi) \cap T = \{0\}\}\right] \\
&\geq \frac{1}{k-1} \left[(l-1) \cdot l + l \cdot (k-l)\right] \\
&= (l^2 - l + lk - l^2)/(k-1) = l \cdot (k-1)/(k-1) = l
\end{aligned}$$

Meanwhile, the RHS is l, so the constraint is satisfied. □

Similar to the rank 1 case, for exact optimality, restrict to when the kernels of the ϕ_i are independent. Again let e_i denote a non-zero smallest Euclidean norm representative of $\ker(\phi_i)$, and let E be the subgroup they span.

Proposition 24. *Suppose the number of maps is equal to k and the kernels are independent. Form the set $S := \{\sum a_i \cdot e_i | a_i \in \mathbb{Z}, 0 \leq a_i \leq \lfloor \frac{M}{k} \rfloor^{\frac{1}{k-1}} - 1\}$. That is, apply Algorithm 1 to the independent elements e_i of subgroup E, with scaling $y_E = 1/(k-1)$ and memory parameter M/k. Then S meets the criteria of Eq. 1.5 and 1.6 for exact optimality.*

Proof. (Sketch). We established that $s_i = 1/(k-1)$ has $1^T s = s_{\text{HBL}}$. Moreover, because the values are evenly distributed, it minimizes γ. Plugging this in, $c_i = 1/k$, $\gamma = (\frac{1}{k})^{k/(k-1)}$.

The remainder follows analagously the argument of rank one maps: show that M/d being rounded induces $1 + o(1)$ relative difference between $\gamma \cdot M^{s_{\text{HBL}}}$ and $|S|$ for Eq. 1.5 to be satisfied, and then quickly confirm Eq. 1.6 holds.

□

6. Acknowledgments

This research is supported in part by NSF grants ACI-1339676, and DOE grants DOE CS-SC0010200, DOE DE-SC008700, DOE AC02-05CH11231, ASPIRE Lab industrial sponsors and affiliates Intel, Google, Hewlett-Packard, Huawei, LGE, NVIDIA, Oracle, and Samsung. Other industrial supporters are Mathworks and Cray. Any opinions, findings, conclusions, or recommendations in this paper are solely those of the authors and does not necessarily reflect the position or the policy of the sponsors.

References

[BCCT] Bennett, A. Carbery, M. Christ, and T. Tao. “Finite bounds for Holder-Brascamp-Lieb multilinear inequalities”. In: *Mathematical Research Letters* 55(4): 647-666 (2010).

[CDKSY13] M. Christ, J. Demmel, N. Knight, T. Scanlon, and K. Yelick. “Communication Lower Bounds and Optimal Algorithms for Programs That Reference Arrays Part 1”. In: *EECS Technical Report* 2013-61 (2013).

[CDKSY15] M. Christ, J. Demmel, N. Knight, T. Scanlon, and K. Yelick. “On Holder-Brascamp-Lieb inequalities for torsion-free discrete Abelian groups”. In: *pre-print* (2015). arXiv: `1510.04190`.

[D] Dedekind. “Uber die von drei Moduln erzeugte Dualgruppe”. In: *Annals of Mathematics* 53 (1900), pp. 371–403.

[DGKSY] M. Driscoll, E. Georganas, P. Koanantakool, E. Solomonik, and K. Yelick. “A communication-optimal N-body algorithm for direct interaction”. In: *Parallel and Distributed Processing* IEEE 27th International Symposium (2013), pp. 1075–1084.

[GGOW] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson. “Algorithmic Aspects of Brascamp-Lieb Inequalities”. In: *pre-print* (2016). arXiv: `1607.06711`.

[KY] P. Koanantakool and K. Yelick. “A Computation- And Communication-Optimal Parallel Direct 3-Body Algorithm”. In: *26th ACM/IEEE Supercomputing Conference* (2014).

[V] S. I. Valdimarsson. “The Brascamp-Lieb Polyhedron”. In: *Canadian Journal of Mathematics* 62 (2010), pp. 870–888.

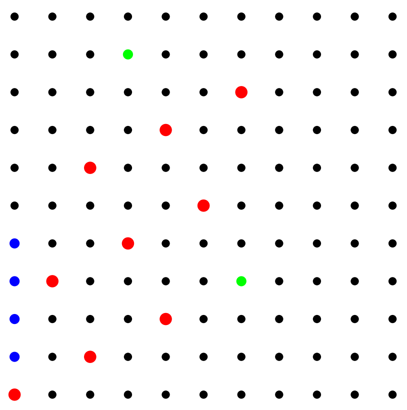

Figure A.1
Red: elements of S
Blue: representatives of other cosets
Green: two starting points of copies of S, extending coverage of the red coset

Appendix A Examples

We will use this appendix to concretely demonstrate certain key points and techniques in the paper. Each example emphasizes different aspects.

A.1 Rank-one maps

Consider $\mathbb{Z}^2$ and maps

$$\phi_1(x, y) = 3x - y$$

$$\phi_2(x, y) = x - 2y$$

The kernels are respectively $\langle e_1 + 3e_2 \rangle$ and $\langle 2e_1 + e_2 \rangle$.

Figure A.1 depicts the tile shape S that is produced by Algorithm 3 when $M = 6$. It also shows a representative of the other 4 cosets of $\mathbb{Z}^2 / \langle e_1 + 3e_2, 2e_2 + e_3 \rangle$. That is, it depicts T_3 of Algorithm 1. T_2 is $\{0\}$ for this example, because $\langle e_1 + 3e_2, 2e_2 + e_3 \rangle$ has the same rank as $\mathbb{Z}^2$. The two green dots correspond to two of the smallest elements of T_1, which accounts for the finite size of M. If we were tiling, copies of S would be translated to start there.

We also alluded to a situation in which infinite data re-use is possible, when the HBL primal LP is infeasible. Consider the computation lattice is $\mathbb{Z}^2$ with a single map $\phi_1(x, y) = x$. The entire tile $\langle e_2 \rangle$ is mapped to 0. So with $M = 1$, we could perform an infinite number of calculations.

A.2 Multiple Tilings and One with non-zero T_2

Consider the following loop nest:

Loop over e_1,e_2,e_3,e_4
    inner loop($A_1[e_1, e_3]$, $A_2[e_2, e_4]$, $A_3[e_1, e_2, e_3 + e_4]$, $A_4[e_1 + e_2, e_3, e_4]$)

This corresponds to the HBL problem in a $d = 4$ dimensional lattice, and linear maps

$$\phi_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \phi_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \phi_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \phi_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

As the lattice is $\mathbb{Z}^4$, we can efficiently calculate the subgroups needed to formulate the primal. They are:

$\langle e_3 - e_4\rangle$, $\langle e_1 - e_2\rangle$, $\langle e_1 - e_2, e_3 - e_4\rangle$, $\langle e_2, e_4\rangle$,
$\langle e_1, e_3\rangle$, $\langle e_2, e_3, e_4\rangle$, $\langle e_1, e_2, e_4\rangle$, $\langle e_1, e_2, e_4\rangle$, $\langle e_1, e_2, e_3\rangle$, $\langle e_1, e_2, e_3, e_4\rangle$

When we solve the primal LP using the computational algebra software Magma, $s = (0, .5, .5, .5)$ so $s_{\text{HBL}} = 1.5$.

Solving the dual LP with Magma, we get a solution supported on subgroups

$$\langle e_1, e_2, e_3, e_4\rangle,\ \langle e_1 - e_2, e_3 - e_4\rangle$$

with dual values of .25 for each. One flag decomposition (Def. 16) of this flag uses subgroups

$$\langle e_1 - e_2, e_3 - e_4\rangle, \langle e_1, e_3\rangle$$

with dual values .5 and .25 respectively. Inputting this to Algorithm 1 in the natural way, the tiling set would be

$$\{a_1\cdot(e_1-e_2)+a_2\cdot(e_3-e_4)+a_3\cdot e_1+a_4\cdot e_3 \,|\, a_i \in \mathbb{Z},\ 0 \le a_1, a_2 \le \lfloor M^{.5}\rfloor-1,\ 0 \le a_3, a_4 \le \lfloor M^{.25}\rfloor-1\}$$

This example illustrates that asymptotically optimal tilings can substantially differ. Checking the following by hand, another asymptotically optimal tiling for this problem defines the tiling set to be

$$\{a_1 \cdot e_1 + a_2 \cdot e_3, +a_3 \cdot (e_2 - e_3) \,|\, a_i \in \mathbb{Z},\ 0 \le a_i \le \lfloor M^{.5}\rfloor - 1\}$$

This shape is fundamentally different than the previous. Notably, direction e_4 is completely ignored. Algorithm 1 for this example outputs $T_2 = \langle e_4\rangle$ to account for the fact that the tile is within a rank 3 subgroup of $\mathbb{Z}^4$. Although in this example there is still an optimal full dimensional tile, the next example uses a rank 7 tile in $\mathbb{Z}^8$ and this is the only optimal tiling we found. Examples exhibiting this behavior are plentiful.

A.3 Need for Flags

We would like to exhibit a situation in which the dual LP excludes a dual vector which yields a valid tiling. This motivates our use of flags to find tilings. Consider the computation lattice $\mathbb{Z}^8$ with maps ϕ_1, ϕ_2, ϕ_3 have kernels given respectively by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Since there are only 3 maps, this is another case in which primal LP can be formulated efficiently. The solution that Magma outputs is is $s = (1, 1, 0)$ so that $s_{\text{HBL}} = 2$. Moreover, the output dual solution it outputs is supported on subgroups

$$\langle e_1 + e_7 - e_8, e_2 + e_8, e_3 + e_7 - e_8, e_5 + e_7 - e_8, e_6 + e_8\rangle, \langle e_1, e_2, e_3, e_6, e_7 + e_8\rangle$$

with dual values of .1 and .3 respectively. Applying one iteration Algorithm 2 and forming a flag decomposition of Def. 16, the new dual solution is supported on subgroups

$$Y_1 := \langle e_1 + 2e_6 + e_7 + e_8, e_3 + 2e_6 + e_7 + e_8, e_2 - e_6\rangle, \; Y_2 := \langle e_1, e_2\rangle, \; Y_3 := \langle e_5, e_7\rangle$$

with dual values .4, .3, .1.

Now for the main point of the example; we calculate $C_1(y)$ as in Eq. 3.2. It turns out $C_3(y) = 1.2$. Indeed,

$$y_{Y_1} \cdot \operatorname{rank}(\phi_3(Y_1)) + y_{Y_2} \cdot \operatorname{rank}(\phi_3(Y_2)) + y_{Y_3} \cdot \operatorname{rank}(\phi_3(Y_3)) = .4 \cdot 2 + .3 \cdot 2 + .1 \cdot 2 = 1.6$$

Proposition 18 is critical here, because $\phi_3(Y_2 + Y_1) = \phi_3(Y_1)$, meaning the $.3 \cdot 2$ term is unnecessary.