Communication Lower Bounds and Optimal Algorithms for Programs that Reference Arrays

James Demmel
UC Berkeley
Math and EECS Depts.

Joint work with
Michael Christ, Nicholas Knight, Thomas Scanlon, Katherine Yelick

Motivation: Why avoid communication?

- Communication = moving data
  - Between levels of memory hierarchy
  - Between processors over network
- Running time of an algorithm is sum of 3 terms:
  - #flops * time_per_flop
  - #words_moved / bandwidth ... communication
  - #messages * latency ... communication
- Time_per_flop $\ll$ 1/bandwidth $\ll$ latency
  - Gaps growing exponentially
- Avoid communication to save time
- Same story for energy: Avoid communication to save energy

Example: Optimal Sequential Matmul

- Naive code
  - for i=*1:n*, for j=*1:n*, for k=*1:n*, $C(i,j) += A(i,k) * B(k,j)$
  - Moves $\Theta(n^3)$ words between cache (size $M < n^2$) and DRAM
- "Blocked" code
  - Write A as $n/b \times n/b$ matrix of $b \times b$ blocks $A[i,j]$
  - Ditto for B, C
  - for i=*1:n/b*, for j=*1:n/b*, for k=*1:n/b*,
    $C[i,j] += A[i,k] * B[k,j]$ ... $b \times b$ matmul
- Thm [Hong,Kung]: Choosing $b \lessapprox (M/3)^{1/2}$ attains lower bound:
  #words_moved $= \Omega(n^3/M^{1/2})$
- Where do $1/2$'s come from?

New Theorem, applied to Matmul

- for i=1:n, for j=1:n, for k=1:n, $C(i,j) += A(i,k) * B(k,j)$
- Record array indices in matrix Δ

$$\Delta = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{matrix} i & j & k \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

- Let $x = [x_i, x_j, x_k]^T$, $\mathbf{1}$ = vector of 1's
- Solve LP: maximize $\mathbf{1}^T x$ such that $\Delta x \le \mathbf{1}$
- Solution: $x = [1/2, 1/2, 1/2]$, $\mathbf{1}^T x = 3/2 \equiv s_{HBL}$
- Thm: #words_moved = $\Omega(n^3/M^{s_{HBL}-1}) = \Omega(n^3/M^{1/2})$.
- Attain by blocking index i by $\Theta(M^{x_i}) = \Theta(M^{1/2})$, ditto for j, k

New Theorem, applied to Direct n-Body

- for *i=1:n*, for *j=1:n*, $F(i) += force(P(i), P(j))$
- Record array indices in matrix Δ

$$\Delta = \begin{array}{c} \\ F \\ P(i) \\ P(j) \end{array} \begin{array}{c} \begin{matrix} i & j \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

- Let $x = [x_i, x_j]^T$, $\mathbf{1}$ = vector of 1's
- Solve LP: maximize $\mathbf{1}^T x$ such that $\Delta x \leq \mathbf{1}$
- Solution: $x = [1, 1]$, $\mathbf{1}^T x = 2 \equiv s_{HBL}$
- Thm: #words_moved $= \Omega(n^2/M^{s_{HBL}-1}) = \Omega(n^2/M^1)$.
- Attain by blocking index i by $\Theta(M^{x_i}) = \Theta(M^1)$, ditto for j

New Theorem, applied to Random Code

- for *i1=1:n*, ... , for *i6=1:n*,
  $A1(i1,i3,i6) += func1(A2(i1,i2,i4), A3(i2,i3,i5), A4(i3,i4,i6))$
  $A5(i2,i6) += func2(A6(i1,i4,i5), A3(i3,i4,i6))$
- Record array indices in 6×6 matrix Δ
  - one column per index $i1$,...,$i6$
  - one row per distinct set of array subscripts $A1$, ..., $A6$
  - $\Delta(i,j) = 1$ if array subscript i has index j, else 0
- Let $x = [x_1, ..., x_6]^T$, $\mathbf{1}$ = vector of 1's
- Solve LP: maximize $\mathbf{1}^T x$ such that $\Delta x \le \mathbf{1}$
- Solution: $x = [2/7, 3/7, 1/7, 2/7, 3/7, 4/7]$, $\mathbf{1}^T x = 15/7 \equiv s_{HBL}$
- Thm: #words_moved $= \Omega(n^6/M^{s_{HBL}-1}) = \Omega(n^6/M^{8/7})$.
- Attained by block sizes $M^{2/7}$, $M^{3/7}$, $M^{1/7}$, $M^{2/7}$, $M^{3/7}$, $M^{4/7}$

Summary of Results (1/3)

- Extend communication lower bound proof from linear algebra to any program with
  - Inner loop iterations indexed by $(i_1, ..., i_d)$
  - Arrays in inner loop subscripted by *linear* functions of indices
  - Ex: $A(i_1, i_2 - i_1, 3i_1 - 4i_2 + 7i_4, ...)$, $B(pntr(i_5 + 6i_6))$, ...
  - Can be dense or sparse, sequential or parallel, ...
- Based on recent generalization of Hölder, Loomis-Whitney, Brascamp-Lieb inequalities by Bennett/Carbery/Christ/Tao
  - Need to count lattice points, not volumes
  - Get linear program with one inequality per subgroup $H \leq \mathbb{Z}^d$
  - Solution of linear program (HBL-LP) is s_{HBL}
  - Thm: #words_moved = Ω(#loop_iterations$/M^{s_{HBL}-1}$)

Summary of Results (2/3)

- Can we write down the lower bound?
  - One inequality per subgroup $H \leq \mathbb{Z}^d$, but still finitely many!
  - Thm (Bad news): Writing down all inequalities in HBL-LP $\iff$ Hilbert's 10th Problem over $\mathbb{Q}$
  - Thm (Good news): Another LP has same solution, is decidable (but expensive, so far)
  - Thm (Better news): "Easy" to write down HBL-LP explicitly in many cases of interest
    - When subscripts are just subsets of indices
    - When #arrays at most 3 (Dedekind)
    - When #loop_indices at most 4
    - When #subscripts $\in \{1, \#loop\_indices - 1\}$
  - Possible class project: implement special cases

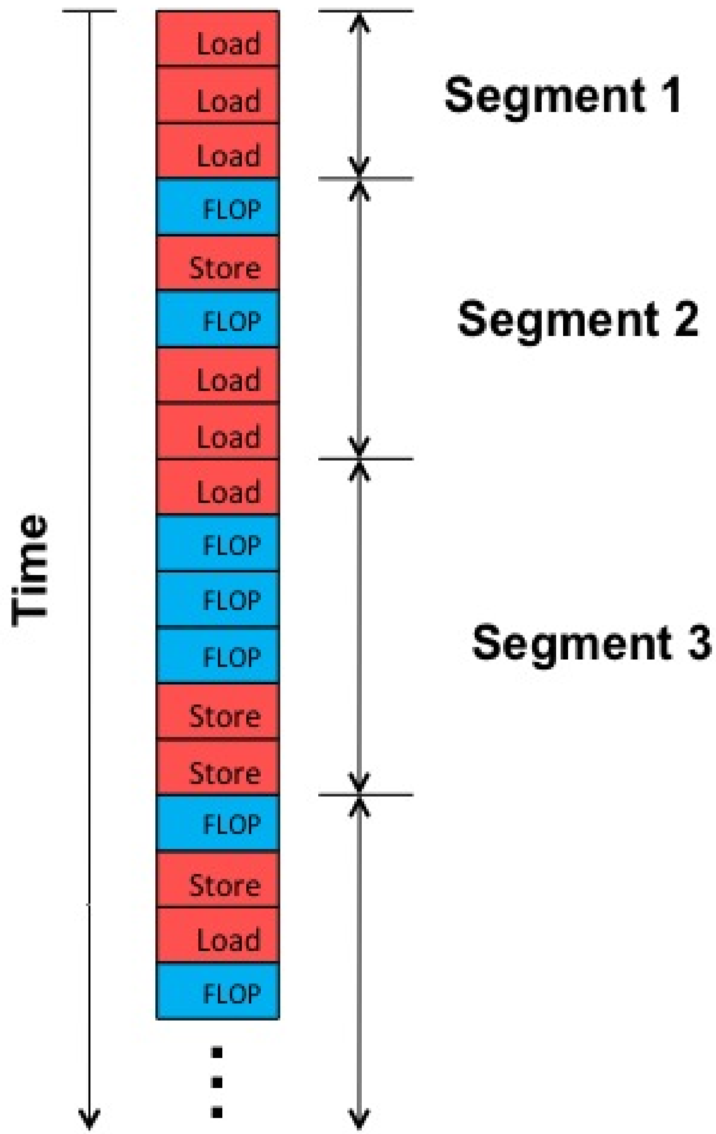
Summary of Results (3/3)

- Can we attain the lower bound?
  - Depends on loop dependencies
  - Best case: none, or reductions (like matmul)
  - Thm: When subscripts are just subsets of indices, the solution x of *dual* HBL-LP tells us the optimal tile sizes M^{x_1},...,M^{x_d}
  - Ex: linear algebra, n-body, "random code", database join, ...
  - Conjecture: always attainable (modulo dependencies)
  - Possible class projects (details later)
    - See if special cases on previous slide attainable
    - Incorporate dependencies into LP, to optimize communication subject to dependencies
    - Incorporate into a real compiler...

Outline

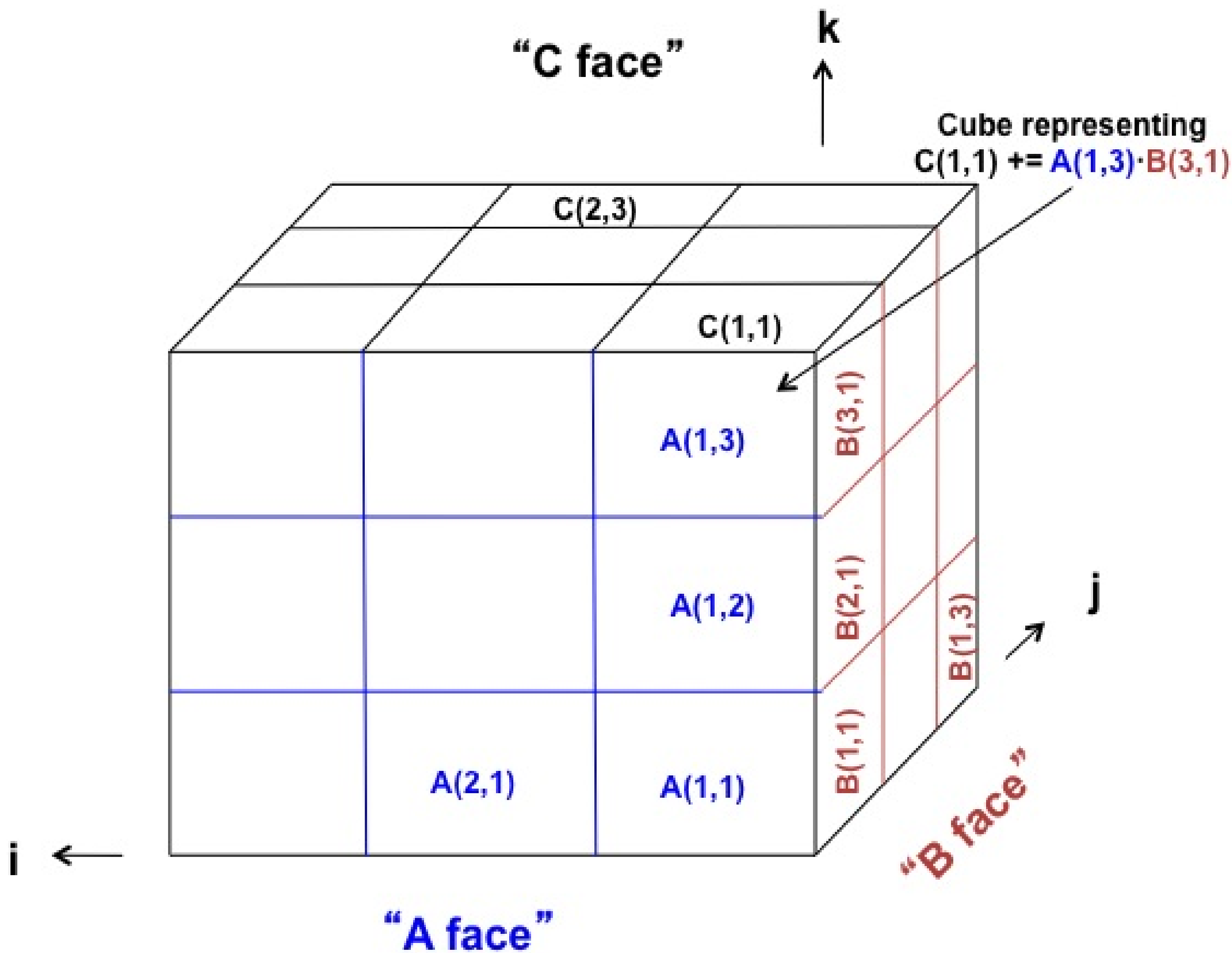
1. Lower bound proof for direct linear algebra using Loomis-Whitney
2. Hölder-Brascamp-Lieb Linear Program (HBL-LP)
   - Continuous case, then discrete case
3. Applying lower bound to more general code
4. Decidability of lower bound
   - Where Hilbert's 10th Problem over $\mathbb{Q}$ arises, how to avoid it
5. Special Case: When subscripts are just subsets of indices
   - Why HBL-LP simpler, why dual tells us optimal algorithm
6. Conclusions and Open Problems

Proof for Direct Linear Algebra (3 Nested Loops)



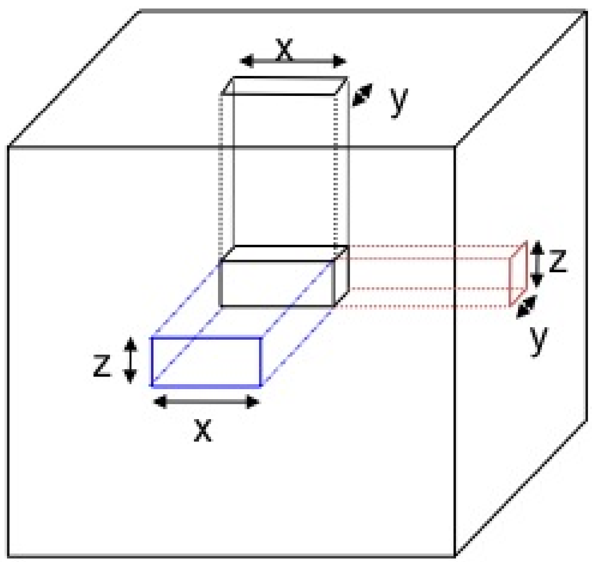
- M = fast memory size
- G = total number of flops
- Break instruction stream into "segments" of M loads/stores
- Data available per segment = 2*M
- Somehow derive upper bound F on #flops possible per segment
- #segments * F ≥ G
- #loads/stores = M * #segments
  ≥ MG/F
- All depends on upper bound F

Geometric Model



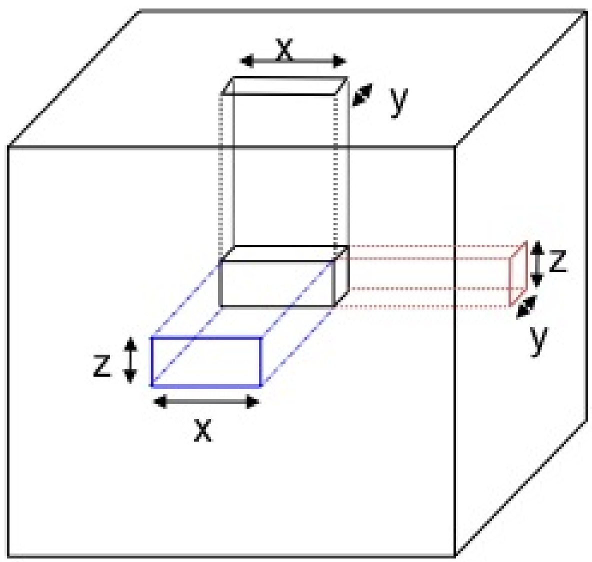
If we have at most 2M “A squares”, 2M “B squares”, and 2M “C squares” on faces, how many cubes can we have?

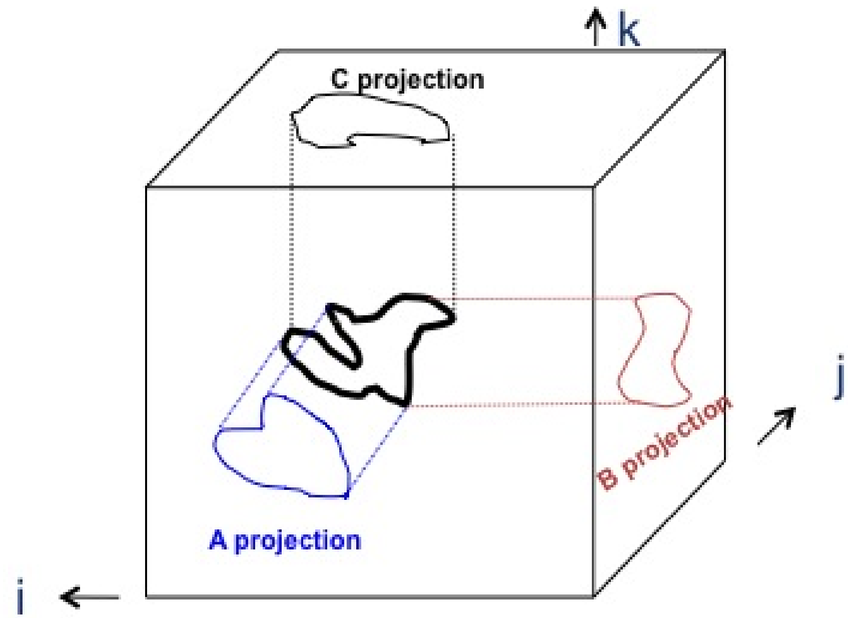
Loomis-Whitney



\# cubes in black box with
side lengths x, y and z
= Volume of black box
= $x \cdot y \cdot z$
= $(xz \cdot zy \cdot yx)^{1/2}$
= (#A□s · #B□s · #C□s $)^{1/2}$

Loomis-Whitney





cubes in black box with side lengths x, y and z
= Volume of black box
= x·y·z
= (xz · zy · yx)$^{1/2}$
= (#A□s · #B□s · #C□s)$^{1/2}$

(i,k) is in A projection if (i,j,k) in 3D set
(j,k) is in B projection if (i,j,k) in 3D set
(i,j) is in C projection if (i,j,k) in 3D set

Thm (Loomis & Whitney, 1949)
cubes in 3D set = Volume of 3D set
≤ (area(A projection) ·
area(B projection) ·
area(C projection)) $^{1/2}$

Summary of Lower Bound Proof for 3 Nested Loops

- M = fast memory size, G = total number of flops
- Break instruction stream into segments of M loads/stores
- $\Longrightarrow 2M$ words of data available during segment
- Use Loomis Whitney to bound F = #multiplies/segment by

$$\begin{aligned} F &\leq (\#\text{A\_entries})^{1/2} \cdot (\#\text{B\_entries})^{1/2} \cdot (\#\text{C\_entries})^{1/2} \\ &\leq (2M)^{3/2} = O(M^{3/2}) \end{aligned}$$

- $F \cdot$#segments$\geq G \Longrightarrow$#segments $\geq G/F$
- #loads/stores $= M\cdot$ #segments $\geq MG/F = \Omega(G/M^{1/2})$
- Result independent of dependencies (so works for LU, etc)
- Result independent of G (so works for sparse, parallel etc)
- Bound decreases with $M \Longrightarrow$ replication may help (2.5D algs)

First Extension Strategy

- Loomis-Whitney $\Longrightarrow$ Hölder-Brascamp-Lieb (HBL)
- Volume of $E \subset \mathbb{R}^3 \Longrightarrow$ Volume of $E \subset \mathbb{R}^d$
- Projections from (i, j, k) to (i, j), (i, k), $(k, j) \Longrightarrow$ any linear projections $\phi_1, ..., \phi_m$
- $\mathrm{vol}(E) \le (\mathrm{area}(E_{ij}))^{1/2} \cdot (\mathrm{area}(E_{ik}))^{1/2} \cdot (\mathrm{area}(E_{jk}))^{1/2} \Longrightarrow$ $\mathrm{vol}(E) \le C \cdot \prod_{i=1}^{m} \mathrm{vol}(\phi_i(E))^{s_i}$

Where do we get exponents s_i and $C < \infty$?

Continuous HBL

Continuous HBL Linear Program (C-HBL-LP):

$$\dim(\mathbb{R}^d) = d = \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(\mathbb{R}^d)) = \sum_{i=1}^{m} s_i \cdot d_i$$

and for all subspaces $H \le \mathbb{R}^d$, $\dim(H) \le \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(H))$

Note: There exist infinitely many H, but only finitely many possible constraints in C-HBL-LP (at most $(d+1)^{m+1}$)

Thm (B/C/C/T): $s_i \ge 0$ satisfy C-HBL-LP *if and only if* $\exists\, C < \infty$ such that for all $f_i : \mathbb{R}^{d_i} \to [0, \infty)$ in L_{1/s_i}

$$\int_{\mathbb{R}^d} \prod_{i=1}^{m} f_i(\phi_i(x))dx \le C \cdot \prod_{i=1}^{m} (\int_{\mathbb{R}^{d_i}} [f_i(y)]^{1/s_i} dy)^{s_i} = C \cdot \prod_{i=1}^{m} \|f_i\|_{1/s_i}$$

Continuous HBL - Special case (1/3)

$$\dim(\mathbb{R}^d) = d = \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(\mathbb{R}^d)) = \sum_{i=1}^{m} s_i \cdot d_i$$

and for all subspaces $H \leq \mathbb{R}^d$, $\dim(H) \leq \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(H))$

Thm (B/C/C/T): $s_i \geq 0$ satisfy C-HBL-LP *if and only if* $\exists\, C < \infty$ such that for all $f_i : \mathbb{R}^{d_i} \to [0, \infty)$ in L_{1/s_i}

$$\int_{\mathbb{R}^d} \prod_{i=1}^{m} f_i(\phi_i(x))dx \leq C \cdot \prod_{i=1}^{m} \|f_i\|_{1/s_i}$$

Hölder's Inequality: Choose all ϕ_i = identity, so $\sum_{i=1}^{m} s_i = 1$

$$\| \prod_{i=1}^{m} f_i(x) \|_1 \leq C \prod_{i=1}^{m} \|f_i\|_{1/s_i}$$ ... can show $C = 1$

Continuous HBL - Special case (2/3)

$$\dim(\mathbb{R}^d) = d = \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(\mathbb{R}^d)) = \sum_{i=1}^{m} s_i \cdot d_i \qquad (*)$$

and for all subspaces $H \le \mathbb{R}^d$, $\dim(H) \le \sum_{i=1}^m s_i \cdot \dim(\phi_i(H))$

Thm (B/C/C/T): $s_i \ge 0$ satisfy C-HBL-LP *if and only if* $\exists\, C < \infty$ such that for all $f_i : \mathbb{R}^{d_i} \to [0, \infty)$ in L_{1/s_i}

$$\int_{\mathbb{R}^d} \prod_{i=1}^{m} f_i(\phi_i(x))dx \le C \cdot \prod_{i=1}^{m} \|f_i\|_{1/s_i}$$

Brascamp-Lieb Inequality: Given only (*), C maximized by $f_i(x) = \exp(-x^T A_i x)$ for some s.p.d. A_i (C could be ∞)

Continuous HBL - Special case (3/3)

$$\dim(\mathbb{R}^d) = d = \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(\mathbb{R}^d)) = \sum_{i=1}^{m} s_i \cdot d_i$$

and for all subspaces $H \leq \mathbb{R}^d$, $\dim(H) \leq \sum_{i=1}^{m} s_i \cdot \dim(\phi_i(H))$

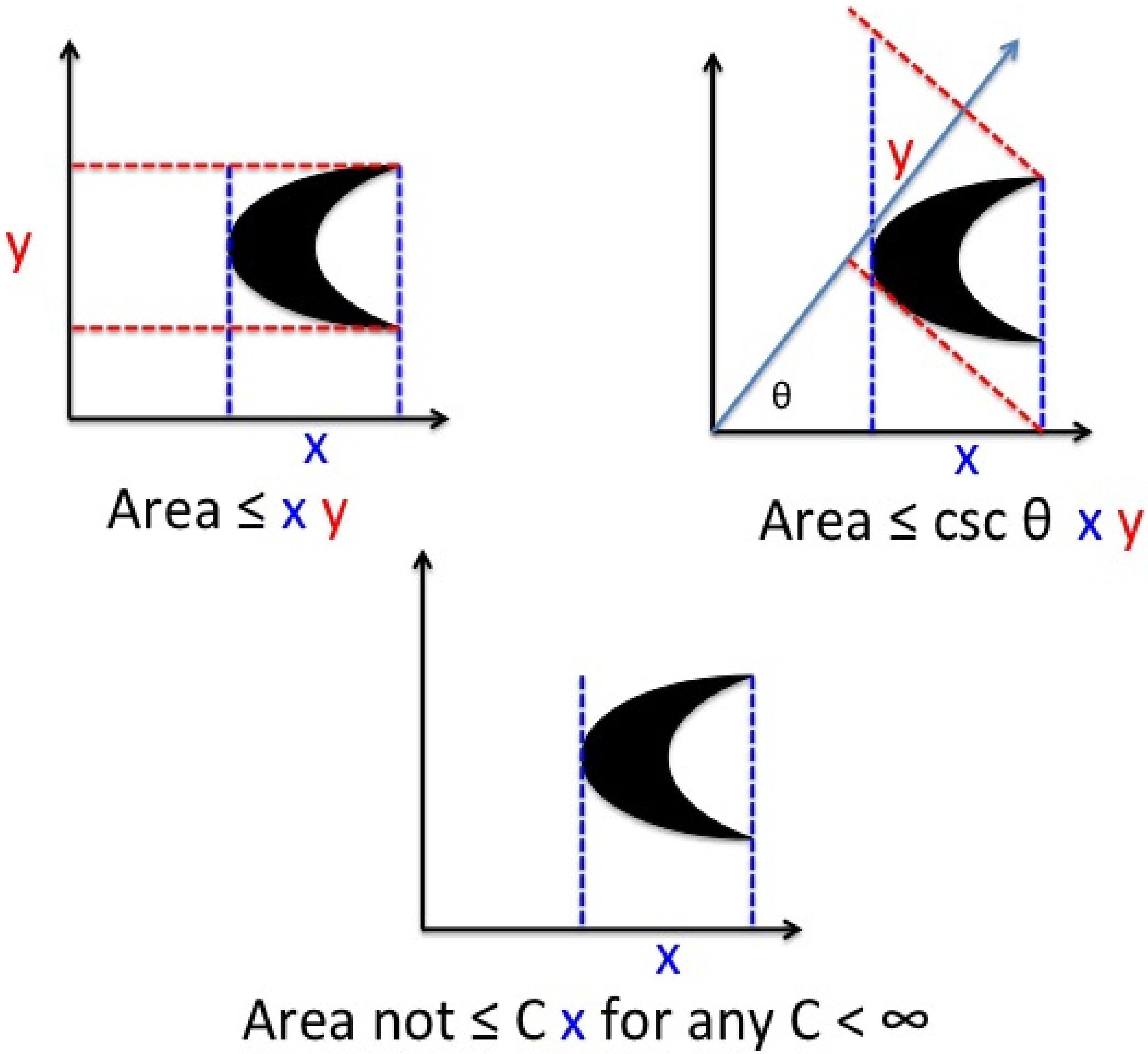
Thm (B/C/C/T): $s_i \geq 0$ satisfy C-HBL-LP *if and only if* $\exists\, C < \infty$ such that for all $f_i : \mathbb{R}^{d_i} \to [0, \infty)$ in L_{1/s_i}

$$\int_{\mathbb{R}^d} \prod_{i=1}^{m} f_i(\phi_i(x))dx \leq C \cdot \prod_{i=1}^{m} \|f_i\|_{1/s_i}$$

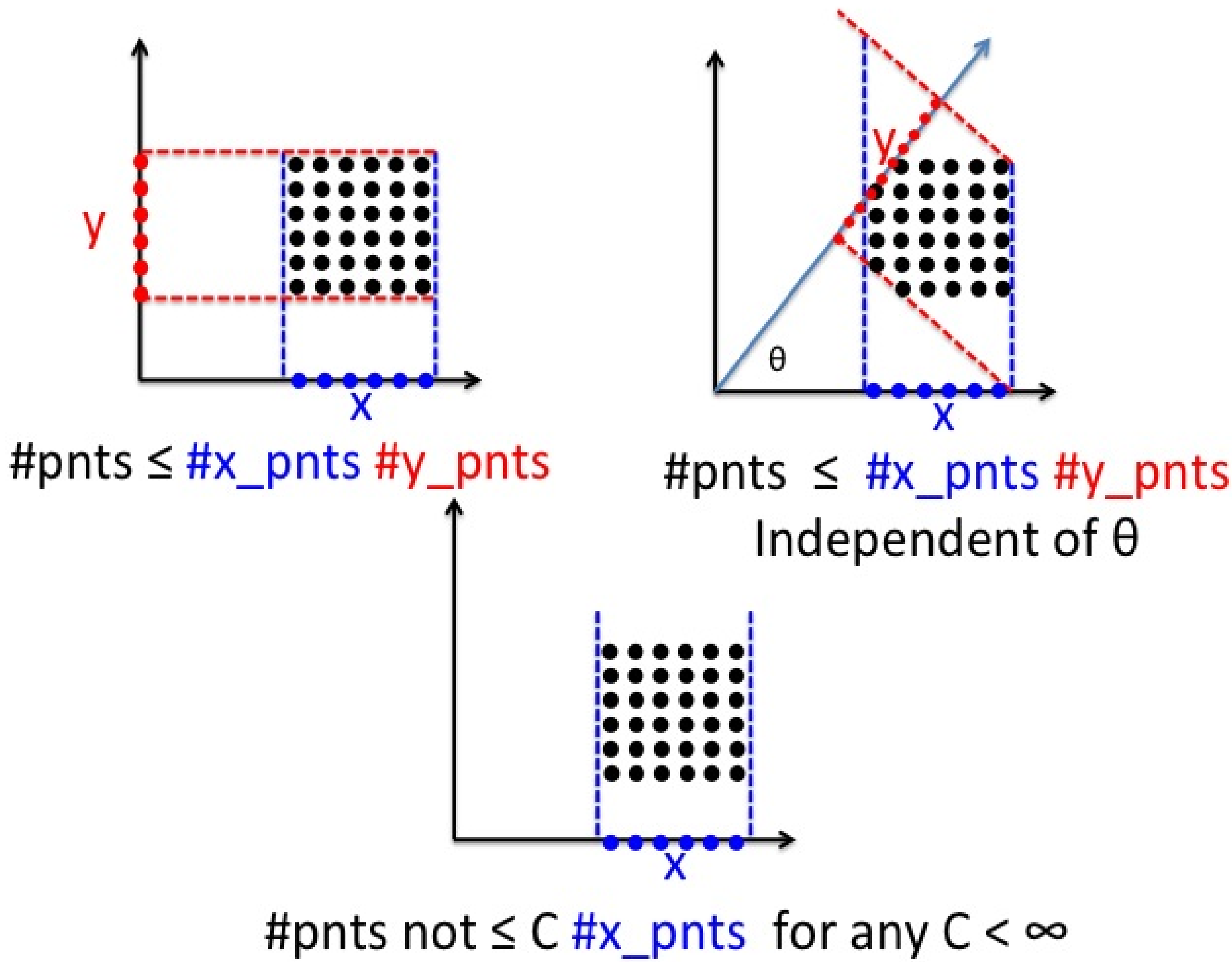
Loomis-Whitney & beyond: Given bounded $E \subset \mathbb{R}^d$, f_i = indicator function of $\phi_i(E)$,

$$\text{vol}(E) \leq C \cdot \prod_{i=1}^{m} (\text{vol}(\phi_i(E)))^{s_i}$$

Illustration of C-HBL-LP



But we want to count lattice points $\equiv$ loop iterations



Second Extension Strategy: Discrete HBL (1/2)

- Count lattice points instead of volumes:
  - Lattice points correspond to loop iterations

$$(i, j, k) \longleftrightarrow C(i, j) += A(i, k) * B(k, j)$$

  - Projected lattice points correspond to array entries

$$(i, j) \longleftrightarrow C(i, j), \quad \text{etc}$$

- Vector space $\mathbb{R}^d \Longrightarrow$ abelian group $\mathbb{Z}^d$ under addition
- Subspaces $H \leq \mathbb{R}^d \Longrightarrow$ subgroups $H \leq \mathbb{Z}^d$
- Linear projection $\phi_i \Longrightarrow$ group homomorphism ϕ_i
- Subspace $\phi_i(H) \Longrightarrow$ subgroup $\phi_i(H)$
- $\dim(H) \Longrightarrow \text{rank}(H)$, $\dim(\phi_i(H)) \Longrightarrow \text{rank}(\phi_i(H))$
- Like C-HBL-LP, but all H, ϕ_i are integer, not real

Second Extension Strategy: Discrete HBL (2/2)

Discrete HBL Linear Program (D-HBL-LP):
for all subgroups $H \le \mathbb{Z}^d$, $\mathrm{rank}(H) \le \sum_{i=1}^m s_i \cdot \mathrm{rank}(\phi_i(H))$

Note: There exist infinitely many H, but only finitely many possible constraints in D-HBL-LP (at most $(d+1)^{m+1}$)

Thm (B/C/<u>C</u>/T): $s_i \ge 0$ satisfy D-HBL-LP *if and only if* for any finite set $E \subset \mathbb{Z}^d$ its cardinality $|E|$ is bounded by

$$|E| \le \prod_{i=1}^{m} |\phi_i(E)|^{s_i} \qquad \ldots \quad C = 1!$$

We want tightest bound when $|\phi_i(E)| \le 2M$, i.e. $|E| \le (2M)^{\sum_{i=1}^m s_i}$
$\Longrightarrow$ Compute $s_{HBL} \equiv \min \sum_{i=1}^m s_i$ subject to D-HBL-LP

Thm: $\#\text{words\_moved} = \Omega(\#\text{iterations}/M^{s_{HBL}-1})$

Some ideas in the proof of Discrete HBL (1/2)

$\forall\, H \le \mathbb{Z}^d$, $\text{rank}(H) \le \sum_{i=1}^{m} s_i{\cdot}\text{rank}(\phi_i(H)) \Longleftrightarrow |E| \le \prod_{i=1}^{m} |\phi_i(E)|^{s_i}$

- Necessity
  - For any $H \le \mathbb{Z}^d$, let E_n be $n \times n \times \cdots \times n$ "brick" in H
  - $|E_n| = \Theta(n^{\text{rank}(H)})$ and $|\phi_i(E_n)| = O(n^{\text{rank}(\phi_i(H))})$

$$\begin{aligned}\Theta(n^{\text{rank}(H)}) &= |E_n| \le \prod_{i=1}^{m} |\phi_i(E_n)|^{s_i} \\ &= O(\prod_{i=1}^{m} n^{s_i\cdot\text{rank}(\phi_i(H))}) = O(n^{\sum_{i=1}^{n} s_i\cdot\text{rank}(\phi_i(H))})\end{aligned}$$

Some ideas in the proof of Discrete HBL (2/2)

$\forall\, H \le \mathbb{Z}^d$, $\mathrm{rank}(H) \le \sum_{i=1}^{m} s_i \cdot \mathrm{rank}(\phi_i(H)) \Longleftrightarrow |E| \le \prod_{i=1}^{m} |\phi_i(E)|^{s_i}$

- Sufficiency (hard part)
  - Suffices to consider extreme points $s = [s_1, ..., s_m]$ of polytope defined by D-HBL-LP
  - Induction over d
  - Def: $H \le \mathbb{Z}^d$ *critical* if $\mathrm{rank}(H) = \sum_{i=1}^{m} s_i \cdot \mathrm{rank}(\phi_i(H))$
  - Given $V \le \mathbb{Z}^d$ and s extreme point, then either $\exists$ critical $\{0\} < H < V$ (induction on H) or $s \in \{0, 1\}^m$

Applying Bounds to More General Code (1/5)

- General model:

$$\begin{array}{l} \text{for all } \mathcal{I} \in \mathcal{Z} \subset \mathbb{Z}^d \text{, in some order} \\ \quad \text{inner\_loop}(\mathcal{I}, A_1(\phi_1(\mathcal{I})), ..., A_m(\phi_m(\mathcal{I}))) \end{array}$$

- Ex: LU inner loop: $A(i,j) = A(i,j) - L(i,k) * U(k,j)$
  - Ok to ignore loop scaling columns of L
  - Ok to overwrite A: $L(i,k) = A(i,k)$ for $i > k$, ditto for U
  - Same idea applies to BLAS, Cholesky, LDL^T, ...
  - Same idea applies to tensor contractions
  - QR, eig, SVD need another idea

Applying Bounds to More General Code (2/5)

- General model:

  for all $\mathcal{I} \in \mathcal{Z} \subset \mathbb{Z}^d$, in some order
  $\quad$ inner_loop($\mathcal{I}, A_1(\phi_1(\mathcal{I})), ..., A_m(\phi_m(\mathcal{I}))$)

- Ex: Computing $B = A^k$ (k odd)
  for $i_1 = 1 : \lfloor k/2 \rfloor$, $C = A \cdot B$, $B = A \cdot C$
- Imperfectly nested loops
- Can't just omit $B = A \cdot C$; infinite data reuse possible, so any lower bound $\propto |\mathcal{Z}|$ must be 0; leads to infeasible HBL-LP
- Solution: *impose reads/writes*: let $\hat{A}[1] = A$, then
  for $i_1 = 2 : k$, $\hat{A}[i_1] = \hat{A}[1] * \hat{A}[i_1 - 1]$
- Apply lower bound to new code, subtract added #reads/writes
- #words_moved $= \Omega(kn^3/M^{1/2} - kn^2) = \Omega(kn^3/M^{1/2})$

Applying Bounds to More General Code (3/5)

- General model:

$$\text{for all } \mathcal{I} \in \mathcal{Z} \subset \mathbb{Z}^d, \text{ in some order}$$
$$\text{inner\_loop}(\mathcal{I}, A_1(\phi_1(\mathcal{I})), ..., A_m(\phi_m(\mathcal{I})))$$

- Ex: Database join

$$\text{for } i_1 = 1 : N_1, \text{ for } i_2 = 1 : N_2$$
$$\text{if predicate}(R(i_1), S(i_2)) = \text{true},$$
$$\text{output}(i_1, i_2) = func(R(i_1), S(i_2))$$

  - Write $\mathcal{Z} = \mathcal{Z}_T \cup \mathcal{Z}_F$, depending on predicate
  - Apply lower bound to $\mathcal{Z}_T$, $\mathcal{Z}_F$ separately, take max
  - #words_moved $= \Omega(\max(|\mathcal{Z}_T|, |\mathcal{Z}|/M))$

Applying Bounds to More General Code (4/5)

- General model:

$$\text{for all } \mathcal{I} \in \mathcal{Z} \subset \mathbb{Z}^d \text{, in some order}$$
$$\quad \text{inner\_loop}(\mathcal{I}, A_1(\phi_1(\mathcal{I})), \ldots, A_m(\phi_m(\mathcal{I})))$$

- Ex: Dense or sparse QR decomposition, using orthogonal transformations
- Not one "algorithm," many variations: un/blocked Givens/Householder, order in which entries zeroed out, ...
- Blocking orth. trans. $\Rightarrow$ imperfectly nested loops
  - Challenge: output of first nest input to second, so need to bound data reuse

Applying Bounds to More General Code (5/5)

- Dense or sparse QR decomposition, continued
- Thm 1: #words_moved = $\Omega(\#\text{flops}/M^{1/2})$ if
  - Blocked Householder with any block sizes
  - One Householder transform per column
- Thm 2: #words_moved = $\Omega(\#\text{flops}/M^{1/2})$ if
  - "Forward Progress": each entry zeroed out once
  - Block size must be 1
- Conjecture: Forward Progress sufficient
- Generalizes to eigenvalue problems

Decidability of the Lower Bound (1/5)

- Recall Continuous HBL-LP: $\dim(\mathbb{R}^d) = d = \sum_{i=1}^m s_i \cdot \dim(\phi_i(\mathbb{R}^d))$ and $\forall H \leq \mathbb{R}^d$, $\dim(H) \leq \sum_{i=1}^m s_i \cdot \dim(\phi_i(H))$
- To write this down, need to solve:
  Given $r_H, r_{H_1}, \ldots, r_{H_m}$, decide if $\exists H \leq \mathbb{R}^d$ s.t.
  $\dim(H) = r_H$, $\dim(\phi_1(H)) = r_{H_1}$,..., $\dim(\phi_m(H)) = r_{H_m}$
- Write H as $d \times d$ matrix
- Write each ϕ_i as $d_i \times d$ matrix
- Express rank conditions by (non)zero constraints on minors
- Tarski-decidable
  - Enough to get upper bound on $s_{HBL} \Longrightarrow$ valid lower bound on communication (possibly too low)

Decidability of the Lower Bound (2/5)

- What about Discrete HBL-LP?
  $\forall H \leq \mathbb{Z}^d$, $\mathrm{rank}(H) \leq \sum_{i=1}^m s_i \cdot \mathrm{rank}(\phi_i(H))$
- To write this down, need to solve:
  Given $r_H, r_{H_1}, ..., r_{H_m}$, decide if $\exists H \leq \mathbb{Z}^d$ s.t.
  $\mathrm{rank}(H) = r_H$, $\mathrm{rank}(\phi_1(H)) = r_{H_1}$,..., $\mathrm{rank}(\phi_m(H)) = r_{H_m}$
- Can encode with minors as before
- Thm: Whether any given system of polynomial equations with rational coefficients has a rational solution or not can be encoded by right choice of $\phi_1, ..., \phi_m$.
- Cor: Being able to write down D-HBL-LP $\iff$
  $\exists$ decision procedure for Hilbert's 10th Problem over $\mathbb{Q}$
  - Over $\mathbb{Q}$ instead of $\mathbb{Z}$ because all conditions homogeneous

Decidability of the Lower Bound (3/5)

- What about Discrete HBL-LP?
  $\forall H \leq \mathbb{Z}^d$, $\text{rank}(H) \leq \sum_{i=1}^m s_i \cdot \text{rank}(\phi_i(H))$
- Constraints define polytope $\mathcal{P}$ in space of $[s_1, ..., s_m] \in \mathbb{R}^m$
- Enough to get any subset of subgroups H defining $\mathcal{P}$
- Let $(H_1, H_2, H_3, ...)$ be any enumeration of all $H \leq \mathbb{Z}^d$
- Let $\mathcal{P}_i$ be polytope defined by $(H_1, ..., H_i)$
- "Simple" decidability algorithm:

$$i = 0, \text{ repeat } i = i + 1 \text{ until } \mathcal{P}_i = \mathcal{P}$$

- Thm: Decidable whether a vertex of $\mathcal{P}_i$ in $\mathcal{P}$
  - Similar induction idea as before
- Better algorithm: which subgroups H to try first?

Decidability of the Lower Bound (4/5)

- Discrete HBL-LP: $\forall H \le \mathbb{Z}^d$, $\mathrm{rank}(H) \le \sum_{i=1}^{m} s_i \cdot \mathrm{rank}(\phi_i(H))$
- Last slide used an enumeration $(H_1, H_2, \ldots)$ of all $H \le \mathbb{Z}^d$
- But most H_j tell us little, unless H_j intersects kernel of ϕ_i, otherwise $\mathrm{rank}(\phi_i(H_j)) = \min(\mathrm{rank}(\phi_i(\mathbb{Z}^d)), \mathrm{rank}(H_j))$,
- So why not just try H_j = kernels of ϕ_i, or "built from them"?
- Def: *Lattice of subgroups of* G, $\mathcal{L}(G_1, \ldots, G_m)$, is formed by taking all possible finite intersections & sums of $\{G_1, \ldots, G_m, \{0\}, G\}$
- Thm: Let $\{\hat{H}_1, \hat{H}_2, \ldots\}$ be enumeration of $\mathcal{L}(\ker(\phi_1), \ldots, \ker(\phi_m))$. Let $\hat{\mathcal{P}}_i$ be polytope defined by $(\hat{H}_1, \ldots, \hat{H}_i)$. Then (potentially much faster) decidability algorithm computes $\mathcal{P}$ correctly:

$$i = 0, \text{ repeat } i = i + 1 \text{ until } \hat{\mathcal{P}}_i = \mathcal{P}$$

Decidability of the Lower Bound (5/5)

- Discrete HBL-LP: $\forall H \le \mathbb{Z}^d$, $\text{rank}(H) \le \sum_{i=1}^{m} s_i \cdot \text{rank}(\phi_i(H))$
- Recall: need to examine polytopes just from subsets of $\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_m))$.
- How big is this lattice? Infinite in general, but ...
- Thm. (Dedekind,1900) When $m = 3$, then $|\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_3))| \le 28$
- Thm. When $\text{rank}(G) \le 4$, then $|\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_m))| = O(m^3)$
- Thm. When all $\text{rank}(\phi_i(G)) \in \{1, \text{rank}(G) - 1\}$, then $|\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_m))| \le 2^m$
- Explicit descriptions of $\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_m))$ in all cases above

Special Case: When subscripts are just subsets of indices (1/3)

- Ex: linear algebra, N-body, database join, ...
  - Matmul: (i, j, k) are indices, subscripts $A(i, k)$, $B(k, j)$, $C(i, j)$
- Much simpler:
  - Easy to write down Discrete HBL-LP to get lower bound
  - Easy to attain lower bound (modulo dependencies): Dual of Discrete HBL-LP gives optimal block sizes
  - Basis of examples at start of talk
- Extends to subsets of unimodular transformations of indices
  - Ex: subsets of $(i, 2i + j, 3i + 2j + k)$

Special Case: When subscripts are just subsets of indices (2/3)

- $i_1, ..., i_d$ be indices, ϕ_1,...,ϕ_m be projections
- Let $\Delta_{j,k} = 1$ if i_k in range of ϕ_j, else 0
- Thm: Let $s = [s_1, ..., s_m]$ minimize $\mathbf{1}^T s \equiv s_{HBL}$ such that $s^T \Delta \geq \mathbf{1}^T$. Then
  #words_moved $= \Omega(\#\text{loop\_iterations}/M^{s_{HBL}-1})$
- Proof idea
  - Constraints $s^T \Delta \geq 1$ are subset of Discrete HBL-LP, for all H spanned by $(0, ..., 0, 1, 0, ..., 0)$ (k-th entry $= 1$)
  - Show this subset implies $\text{rank}(H) \leq \sum_{j=1}^{m} s_j \text{rank}(\phi_j(H))$ for all $H \leq \mathbb{Z}^d$

Special Case: When subscripts are just subsets of indices (3/3)

- $i_1, ..., i_d$ be indices, $\phi_1,...,\phi_m$ be projections
- Let $\Delta_{j,k} = 1$ if i_k in range of ϕ_j, else 0
- Dual LP: Let $x = [x_1, ..., x_d]$ maximize $\mathbf{1}^T x \equiv s_{HBL}$ such that $\Delta x \leq \mathbf{1}^T$.
- Thm: The solution x of the Dual LP gives the optimal block sizes to minimize communication: i_k blocked by M^{x_k}
- Proof idea
  - Each constraint in $\Delta x \leq \mathbf{1}$ bounds number of entries of each array by M
  - $\mathbf{1}^T x = s_{HBL}$ says number of inner loop iterations per block is $M^{s_{HBL}}$.
- Extends to parallel case, "n.5D" algorithms

Conclusions, possible class projects/open problems/theses/...

- Possible to derive decidable communication lower bounds for many widely used algorithms that access arrays
- Possible to achieve these bounds in many cases, leading to faster algorithms
- Possible projects/open problems/etc:
  - Implement analyses to compute lower bounds
  - Conjecture: Dual LP gives tiling for optimal algorithm in (special) cases based on $\mathcal{L}(\ker(\phi_1), ..., \ker(\phi_m))$.
  - Compilers use "polyhedral analysis" to characterize dependencies; can this be combined with our optimal (polyhedral) tilings to minimize communication subject to dependencies?
  - Implement well-understood special cases in compiler, generate optimal code, ...

Key to Success

Key to Success

Don't Communic...