Journal of Parallel & Distributed Computing, 42(1):42--59, 1997

Communication-Minimal Tiling of Uniform Dependence Loops

Jingling Xue

Department of Mathematics, Statistics and Computer Science
University of New England
Armidale, NSW 2351, Australia

Running head: Communication-Minimal Tiling

Mailing address:

Dr Jingling Xue
Department of Mathematics, Statistics and Computer Science
University of New England
Armidale 2351, Australia

Tel: +61 67 733149
Fax: +61 67 733312
Email: xue@cs.une.edu.au

Abstract. Tiling is a loop transformation that a compiler uses to create automatically blocked algorithms in order to improve the benefits of the memory hierarchy and reduce the communication overhead between processors. Motivated by existing results, this paper presents a conceptually simple approach to finding tilings with a minimal amount of communication between tiles. The development of almost all results is based primarily on the inequality of arithmetic and geometric means and the concept of extremal rays from convex cones. The key insight is that a tiling that is communication-minimal must induce the same amount of communication through all faces of a tile, which restricts the search space for optimal tilings to those tiling matrices whose rows are all extremal rays in a cone. For nested loops with several special forms of dependences, closed-form optimal tilings are derived. In the general case, a procedure is given that always returns optimal tilings. An efficient implementation of the procedure, along with experimental results, is presented. A detailed comparison of this work with some existing results is provided.

List of Symbols

$\mathbb{Z}$	set of integers
$\mathbb{Q}$	set of rationals
$\|\| \cdot \|\|$	Euclidean norm
$\mathcal{I}$	the identity matrix
A^T	the transpose of A
diag$(a_1, \cdots, a_n)$	diagonal matrix
$\det(A)$	determinant of A
$x \times y$	vector product of x and y
H	tiling matrix
P	inverse of H
$\mathcal{V}_{\text{comp}}(H)$	computation volume of a tile induced by H
$\mathcal{V}_{\text{comm}}(H)$	communication volume of a tile induced by H
D	dependence matrix
$\tilde{D}$	a matrix obtained from dependence matrix D
$\mathcal{C}(D)$	dependence cone
$\mathcal{C}(h)$	tiling cone
cone$(x_1, \cdots, x_m)$	cone generated by $x_1, \cdots, x_m$

1 Introduction

Studies have shown that blocked algorithms can improve the performance of parallel computers with a memory hierarchy [7, 8]. A block is a subarray of data and usually exhibits a high degree of data reuse, allowing better register, cache and memory hierarchy performance.

Tiling is a loop transformation that a compiler uses to automatically create blocked algorithms [11, 22]. Tiling divides the iteration space into blocks or *tiles* of the same size and shape and traverses the tiles to cover the entire iteration space. To improve cache locality of a loop nest, the compiler can find tiles so that a tile is small enough for cache to capture the available temporal reuse, improving the benefits of the memory hierarchy [4, 13, 17, 23],

Tiling is also a good paradigm for parallel computers with distributed memory. In these multicomputers, the relatively high communication startup cost makes frequent communication very expensive. Tiling can be used to reduce the communication overhead between processors [3, 14, 15, 16]; loop iterations are grouped into tiles, and communication takes place per each tile instead of per each iteration, so that communication overhead is reduced.

This paper is restricted to tightly nested loops with constant dependences, known as *uniform dependence algorithms*. They have the characteristic that data dependences between their computations can be represented by a finite set of integer vectors, known as *dependence* or *distance vectors*. The *iteration space* of a loop nest is a discrete bounded Cartesian space defined by the loop limits of the program. For the purposes of this paper, knowing the dependence information in a program suffices. So an n-deep loop nest is identified by a *dependence matrix* $D = [d_1, \cdots, d_m] \in \mathbb{Z}^{n\times m}$ whose columns are all m dependences vectors in the program. For a sequential program, all dependence vectors are lexicographically positive.

We assume that $D \in \mathbb{Z}^{n\times m}$ has full row rank, implying that $m \geqslant n$. Otherwise, we can always transform an n-deep loop nest into an $(n+\rho)$-deep loop nest consisting of n outer doall loops — some are trivial — and ρ inner sequential loops, where ρ is the row rank of D [1]. The inner ρ loops, having a dependence matrix with full row rank, can be tiled in the normal manner.

Example 1 In the following uniform dependence algorithm:

```
do i = 0, N
   do j = 0, N
      A(i, j) = A(i − 1, j) + A(i − 1, j − 1)
```

The iteration space is a parallelogram: $\{(i, j) \mid 0 \leqslant i, j \leqslant N\}$. The dependence matrix is:

$$D = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

A recent paper described an inspiring approach to finding tilings that minimise the communication volume of a tile when the computation volume or size of a tile is fixed [3]. That approach finds optimal tilings by first determining the shape of a tile and then scaling all sides of a tile by the same constant factor to obtain a tile of an appropriate size. The major results of that work were summarised in [3, Lemma 9 and Theorem 10], which are technically involved and do not seem to lead to an intuitive geometric interpretation. In addition, closed-form optimal tilings in several important special cases were not detailed, and the possible existence of other (infinitely many) optimal tilings other than those found in [3] in the general case was not studied. Finally, the cost of finding optimal tilings was not evaluated.

This paper recasts and extends that work and provides new insights into the problem of finding communication-minimal tilings. Using a different formulation for finding optimal tilings, we are able to develop all results in this paper in a conceptually simpler framework based primarily on the inequality of arithmetic and geometric means and the concept of extremal rays from convex cones. That inequality is the basis for establishing Theorem 1 and Lemma 2 in this paper, the keystones on which all other results rest. One important result is that a tiling that is communication-minimal must induce the same amount of communication (not the same surface area as in [17]) on all faces of a tile. This is the deep reason why the search space for optimal tilings can be restricted to a finite set of matrices whose rows are all extremal rays in a cone. By classifying programs into individual cases in terms of their dependence structures, we find optimal tilings progressively so that each case is solved based on the preceding one. In several important and commonly occurring special cases, the unique closed-form optimal tilings are derived. In the general case, a necessary condition is provided for a tiling to be optimal, requiring that all its row vectors to be contained in the faces of a cone. A necessary and sufficient condition is given for a program to have infinitely many optimal tilings. The existence of infinitely many optimal tilings is parallel to the (degenerate) case when a linear programming problem assumes its optimum at infinitely many solutions. But, just like that a linear programming problem always assumes its (finite) optimum at an extremal point of its solution space, it is also possible to select an optimal tiling from a finite set of matrices whose rows are all extremal rays in a cone. A procedure is given that always returns all these extremal-ray optimal tilings. An efficient implementation of this procedure, along with experimental results, is also provided. Frequently, the simplest interpretation of an algebraic result is in terms of a geometric setting. Where appropriate, some geometric insights behind optimal tilings are explained.

The plan of the paper is as follows. Section 2 introduces the terminology and notations used in the paper. Section 3 discusses tiling as a loop transformation. Section 4 characterises the computation and communication volumes of a tile. In Section 5, the problem of finding communication-minimal tilings is formulated as a combinatorial problem. Several concepts from higher mathematics and convex cones are introduced. We derive optimal tilings by distinguishing programs according to their data dependence structures. We first consider programs with several special forms of data dependences, and in each special case we present the unique closed-form optimal tiling. We then discuss the general case when D is an arbitrary full row matrix. Section 6 contains a procedure for finding optimal tilings and discusses the performance results of the procedure in finding optimal tilings. Based on the framework developed in this paper, Section 7 compares this work with the related work. Section 8 concludes the paper by describing some future work.

2 Notation and Terminology

$\mathbb{Z}$ and $\mathbb{Q}$ denote the set of integers and rationals, respectively. All relational operators on two vectors are component-wise. For example, if x and y are two vectors, then $x \geqslant y$ means that every component of x is greater than or equal to the corresponding component of y. The dimensions of vectors and whether they are row or column vectors are implied by the context in which they are used. The symbol $\mathcal{I}$ denotes the identity matrix. The notation diag$(a_1, \cdots, a_n)$ denotes the square diagonal matrix with numbers $a_1, \ldots, a_n$ on its main diagonal. The transpose of a matrix A is denoted by A^{T}. If x and y are two vectors, $x \cdot y$ (or xy) and $x \times y$ denote the dot product and vector product of the two vectors, respectively. We use $\| \cdot \|$ for the *Euclidean norm*, i.e., $\| x \| = \sqrt{x^{\mathrm{T}} x}$. If $b_1, \cdots, b_n$ are column vectors, and B is the square matrix with columns $b_1, \cdots, b_n$, then we have the *Hadamard inequality* [18, p. 7]:

$$|\det(B)| \quad \leqslant \quad \| b_1 \| \times \cdots \times \| b_n \|$$

where the sign of equality holds if and only if $b_1, \cdots, b_n$ are mutually orthogonal.

We write $\lceil x \rceil$ for the *ceiling* of x and $\lfloor x \rfloor$ for the *floor* of x. If x is an element of a set S, the notation $x \in S$ is used, and this notation is abused to indicate that a column vector x is a column of a matrix M, i.e., $x \in M$.

3 Iteration Space Tiling

This section discusses tiling as a loop transformation introduced in [11, 23, 24]. Tiling decomposes an n-dimensional loop nest into a $2n$-dimensional loop nest where the outer n *tile loops* step between tiles and the inner n *element loops* step the points within a tile. Figure 1(a) shows a 2×2

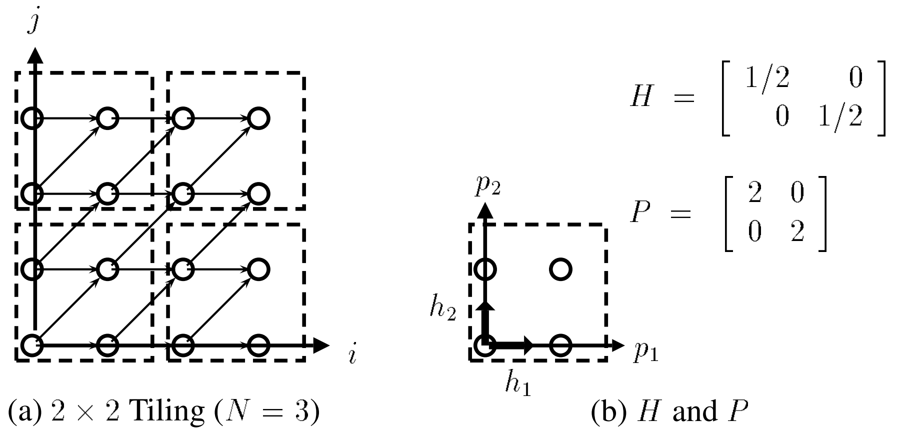


Figure 1: A parallelogram tiling of the double loop in Example 1.

parallelogram tiling of the double loop in Example 1, where the tiled program is as follows:

```
do ii = 0, N, 2
   do jj = 0, N, 2
      do i = ii, min(ii + 1, N)
         do j = jj, min(jj + 1, N)
            A(i, j) = A(i - 1, j) + A(i - 1, j - 1)
```

In general, tiling divides an n-dimensional iteration space into n-dimensional parallelepiped tiles of the same size and shape, and traverses the tiles to cover the entire iteration space. Since all tiles are identical by translation, a tiling transformation can be defined either by the normal vectors to its n faces or by the edge vectors of its n edges emitting from the tile origin. Let H be the *tiling matrix* whose rows are the normal vectors of the n faces of a tile. Let P be the *clustering matrix* whose columns are the edge vectors of a tile. Figure 1(b) shows the H and P for the parallelogram tiling shown in Figure 1(a).

A tiling transformation is defined as a one-to-one mapping from $\mathbb{Z}^n$ to $\mathbb{Z}^{2n}$:

$$I \quad \longmapsto \quad \begin{pmatrix} \lfloor HI \rfloor \\ I - H^{-1}\lfloor HI \rfloor \end{pmatrix}, \quad \text{where } 0 \leqslant HI - \lfloor HI \rfloor < 1$$

where the first component $\lfloor HI \rfloor$ identifies the tile that I belongs to and the second component $I - H^{-1}\lfloor HI \rfloor$ gives the index of I within the tile relative to the tile origin. There are several restrictions that H must satisfy. H is nonsingular so that a tile has a bounded number of points. $P = H^{-1}$ must be integral so that all tiles contain the same number of integer points (identical by translation in $\mathbb{Z}^n$). H must also satisfy a so-called *atomic tiles constraint*. Each tile is an atomic unit of work to be scheduled on a processor. Once a tile is scheduled, it runs to completion without preemption. A tile is executed only if all dependence constraints for that tile have been satisfied, implying that there must not exist any cyclic dependences on the outer n tile loops. In [11], $HD \geqslant 0$ was given as a sufficient condition for enforcing the atomic tiles constraint; it also preserves the dependences of the original program. In a recent paper [25], we presented a necessary and sufficient condition, showed its equivalence with $HD \geqslant 0$ in tiling uniform dependence loops and discussed its implications on dependence abstractions suitable for tiling and its impact on tiling nested loops that are almost uniform.

Some discussions on generating the tiled program for a tiling transformation can be found in [11, 25].

4 Computation and Communication Volumes

This section discusses how to calculate the computation volume and communication volume induced by a tile, providing the basis for formulating the problem of finding communication-minimal tilings in the following section.

The number of integer points or loop iterations contained in a tile is called its *computation volume*. If $P = H^{-1}$ is integral, the computation volume of a tile is given precisely by:

$$\mathcal{V}_{\text{comp}}(H) = \frac{1}{|\det(H)|} = |\det(P)|$$

The communication volume of a tile is defined as the amount of data that must be communicated before a tile can be initiated in a processor. It is measured by a sequence of three approximations [3, 17]. The idea will be illustrated using Figure 2.

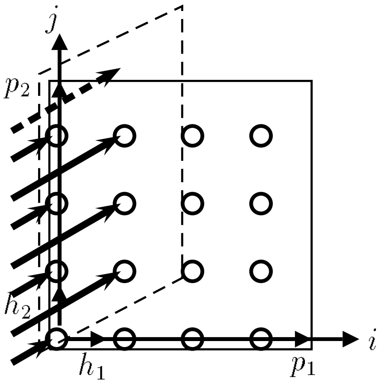
(a) First, the communication volume of a tile is first approximated as the number of dependences that cross the tile boundaries and sink inside the tile. This is an over-approximation because two dependences originating from the same source are counted twice rather than once.

(b) Next, (a) is relaxed by measuring the communication volume of a tile as the number of dependences that cross the tile boundaries. Therefore, a dependence that crosses the tile boundary but does not sink inside the tile will be regarded as contributing to the communication volume of the tile. The dashed dependence in Figure 2 is one such example.

(c) Finally, using (b), we calculate *separately* the communication volumes going through the faces of a tile and then add them up. This over-approximates (b) since a dependence that touches the intersection of several faces is counted multiple times. As an example, in Figure 2, the dependence sinking at the origin will be counted twice, once through each of the two faces touching the origin.

The number of dependences crossing a face can now be calculated as follows. If d is a dependence vector, the number of dependences induced by d through the face h_1 or $p_2 \times \cdots \times p_n$ is equal to the volume of the parallelepiped subtended by $d, p_2, \cdots, p_n$, i.e.,

$$|\det([d, p_2, \cdots, p_n])| = |d \cdot (p_2 \times \cdots \times p_n)| = |\det(P)(h_1 \cdot d)| = \frac{1}{|\det(H)|} h_1 \cdot d$$



$$d = (2,1) \qquad H = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix} \qquad P = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

Figure 2: Approximation of the amount of communication induced by d through the face h_1. The solid box depicts the tile at the origin. The dashed box depicts the parallelogram subtended by d and p_2, whose volume is $\det([d, p_2])$=8, which measures the 8 dependences d crossing the face h_1, of which the one depicted in the dashed arrow sinks outside the tile.

Here, we assume that $h_1 \cdot d \geqslant 0$ because we will enforce the atomic tiles constraint $HD \geqslant 0$ later on. Hence, the communication volume induced by all dependence vectors through the face h_1 is $\frac{1}{|\det(H)|} \sum_{j=1}^{m} h_1 \cdot d_j$. Finally, the communication volume for a tile is the sum of the communication volumes through all faces of the tile:

$$\mathcal{V}_{\mathrm{comm}}(H) \;=\; \frac{1}{|\det(H)|} \sum_{i=1}^{n} \sum_{j=1}^{m} h_i \cdot d_j \;=\; \frac{1}{|\det(H)|} \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{j=1}^{m} h_{i,k} d_{k,j} \tag{1}$$

This formula is a good approximation of the communication volume of a tile. This is because, in practice, the tile size is sufficiently larger than the magnitudes of dependence vectors.

5 Communication-Minimal Tilings

Given the computation volume of a tile as a design parameter ν, this section finds tilings that yield the smallest communication volume for a tile. Formally, we provide optimal solutions to the following optimisation problem:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{V}_{\mathrm{comm}}(H) = \frac{1}{|\det(H)|} \sum_{i=1}^{n} \sum_{j=1}^{m} (HD)_{i,j} \\ \textbf{Subject to} & \mathcal{V}_{\mathrm{comp}}(H) \;=\; \frac{1}{|\det(H)|} = \nu \\ & HD \geqslant 0 \end{array} \tag{2}$$

Here, $HD \geqslant 0$ preserves the atomic tiles constraint, and the problem formulation itself implies that H must be nonsingular. Section 6.4 discusses how to ensure that H^{-1} is integral.

It is clear that we can simplify (2) by making the objective function linear:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n} \sum_{j=1}^{m} (HD)_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & HD \geqslant 0 \end{array} \tag{3}$$

The optimal solutions to both formulations satisfy:

$$\mathcal{V}_{\text{comm}}(H) \;=\; \frac{\mathcal{F}(H)}{|\det(H)|}$$

Note that an optimal tiling will be parameterised by the computation volume ν, and therefore represents a family of optimal solutions for different computation volumes.

In the rest of this section, we provide solutions to (3). At the first glance, this problem is a difficult combinatorial problem. We will show that the problem can be solved analytically based primarily on the inequality of arithmetic and geometric means and the the concept of extremal rays from convex cones. In fact, that inequality will be used to establish almost all results in the paper.

5.1 Background

Lemma 1 **(The Inequality of the Arithmetic and Geometric Means [2, p. 4])** *For any nonnegative numbers $x_1, \cdots, x_n$, we have*

$$(x_1 \times \cdots \times x_n)^{\frac{1}{n}} \;\leqslant\; \frac{x_1 + \cdots + x_n}{n}$$

The sign of equality holds if and only if $x_1 = \cdots = x_n$.

We shall also make use of several basic concepts from convex cones [18, p. 87]. A nonempty set C in Euclidean space is called a convex cone if $\lambda x + \mu y \in C$ whenever $x, y \in C$ and $\lambda, \mu \geqslant 0$. A cone that is *finitely generated* by the vectors $x_1, \cdots, x_m$ is the set:

$$\text{cone}(x_1, \cdots, x_m) \;=\; \{\lambda_1 x_1 + \cdots + \lambda_m x_m \mid \lambda_1, \cdots, \lambda_m \geqslant 0\}$$

That is, the cone consists of all vectors that are nonnegative linear combinations of $x_1, \cdots, x_m$. A cone that is *polyhedral* is the intersection of finitely many linear half spaces:

$$\{x \mid Ax \geqslant 0\}$$

for some matrix A. A convex cone is polyhedral if and only if it is finitely generated. Therefore, a convex cone can be represented in two different forms.

The concept of an extremal ray is introduced next. Since all dependence vectors are lexicographically positive, the dependence cone is a pointed cone with the origin as its apex. The *extremal rays* in a pointed cone are just the edges of the cone. (Formally, a vector r in a cone is an *extremal ray* if there do not exist two linearly independent vectors r_1 and r_2 in the cone such that $r = \frac{1}{2}r_1 + \frac{1}{2}r_2$.)

Two cones are frequently used in the literature on tiling. All dependence vectors in the dependence matrix D generate a cone, called the *dependence cone* [26]:

$$\mathcal{C}(D) \;=\; \text{cone}(d_1, \cdots, d_m)$$

Let h be a row vector in H. Then all feasible *tiling vectors* h are contained in a cone, called the *tiling cone*, in this paper:

$$\mathcal{C}(h) \;=\; \{h \mid hD \geqslant 0\}$$

When constructing optimal tilings in the general case (to be described in Section 5.6), we shall need to know the extremal rays of the tiling cone. These rays can be constructed from the dependence cone, and the construction relies on the duality between the two cones.

Let $d'_1, \cdots, d'_q$ be the q extremal rays of the dependence cone and $D' = [d'_1, \cdots, d'_q]$. Let $r_1, \cdots, r_p$ be the p extremal rays of the tiling cone and $T = [r_1, \cdots, r_p]$. These two cones are related to each other in the following way:

$$\begin{array}{rcl} \mathcal{C}(D) & = & \text{cone}(d_1, \cdots, d_m) \;=\; \text{cone}(d'_1, \cdots, d'_q) \;=\; \{h \mid hT \geqslant 0\} \\ \mathcal{C}(h) & = & \{h \mid hD \geqslant 0\} \;=\; \{h \mid hD' \geqslant 0\} \;=\; \text{cone}(r_1, \cdots, r_p) \end{array} \tag{4}$$

The extremal rays of the tiling cone are the normals to the faces of the dependence cone, and vice versa. The dependence cone has q extremal rays (or edges) and p faces. Dually, the tiling cone has p extremal rays (or edges) and q faces. In addition, p and q satisfy the following properties:

(a) If D has full rank, $p \geqslant n$ and $q \geqslant n$.

(b) If $p = n$, then $q = n$ and vice versa. This is because $D' = (T^{-1})^T$.

(c) Following from both (a) and (b), we know that $p > n$ if and only if $q > n$.

As shown later in the section, we are able to derive the unique closed-form optimal tiling in the special case when $p = q = n$ and have to work harder to find optimal tilings in the general case.

The extremal rays of the tiling cone can be constructed from the faces of the dependence cone as follows [3, 17]. Every set of $n-1$ linearly independent dependence vectors $x_1, \cdots, x_{n-1}$ in D potentially defines a face of the dependence cone. Let r be the normal to the hyperplane spanned by these $n-1$ vectors. One solution is $r = x_1 \times \cdots \times x_{n-1}$. If $rd \geqslant 0$ for all $d \in D$, then r is an extremal ray; or if $rd \leqslant 0$ for all $d \in D$, then $-r$ is an extremal ray; otherwise $x_1 \times \cdots \times x_{n-1}$ does not define a face for the dependence cone. Section 6 describes an efficient approach to constructing the extremal rays of the tiling cone.

Example 2 Consider the dependence matrix:

$$D \;=\; [d_1, d_2, d_3] \;=\; \left[\begin{array}{ccc} 2 & 0 & 2 \\ 1 & 1 & 2 \end{array}\right]$$

The dependence cone has two extremal rays d_1 and d_2. Note that $d_3 = d_1 + d_2$ is not a ray. The tiling cone has two extremal rays $r_1 = (1, 0)$ and $r_2 = (-1, 2)$. Figure 3 depicts both the dependence and tiling cones, which are obtained from (4) as follows:

$$\begin{aligned}
\mathcal{C}(D) &= \text{cone}(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}) = \{\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \mid h_1 \geqslant 0, -h_1 + 2h_2 \geqslant 0\} \\
\mathcal{C}(h) &= \{\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \mid 2h_1 + h_2 \geqslant 0, h_2 \geqslant 0\} = \text{cone}(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix})
\end{aligned}$$

Here, r_1 and r_2 are both the two extremal rays of the tiling cone and the normals to the two faces of the dependence cone. Dually, d_1 and d_2 are both the two extremal rays of the dependence cone and the normals to the two faces of the tiling cone.

For 2-deep and 3-deep loop nests, the dependence (tiling) cone always has the same number of edges as the number of faces. This is not true in the general case, as illustrated by the following example. This example also illustrates further the duality of the dependence and tiling cones.

Example 3 Consider the dependence matrix:

$$D = \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 2 & 1 & 1 & 1 & 0 \\ 2 & -2 & 2 & 2 & 1 \\ -2 & 0 & -1 & 1 & 2 \\ -1 & -1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

The dependence cone has 5 extremal rays, which are the 5 columns of D. Using the tiling procedure in Figure 10, we obtain the 6 extremal rays for the tiling cone:

$$T = \begin{array}{c} \begin{array}{cccccc} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{array} \\ \begin{bmatrix} 4 & 8 & 5 & 1 & 5 & 7 \\ 3 & -3 & -3 & 5 & -1 & -4 \\ 8 & -2 & 1 & 2 & -3 & 1 \\ -2 & 14 & 2 & -9 & 7 & 2 \end{bmatrix} \end{array}$$

According to (4), the dependence cone has 5 edges and 6 faces (whose normals are the columns of T), and the tiling cone has 6 edges and 5 faces (whose normals are the 5 columns of D).

In the rest of this section, we focus on finding optimal solutions to the optimisation problem (3). We distinguish programs in terms of their dependence structures. We first describe closed-form optimal tilings for programs with several special forms of dependence matrices. *All these special cases have one thing in common: the dependence cone in each case has exactly n edges.* We then address the problem of finding optimal tilings in the general case.

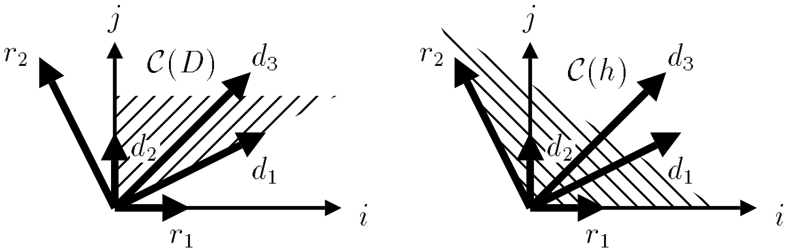


Figure 3: Dependence and tiling cones for Example 2.

5.2 D Is the Identity Matrix

If D is the identity matrix, the optimisation problem (3) becomes:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n} \sum_{j=1}^{m} h_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & H \geqslant 0 \end{array}$$

The optimal solution is found analytically based on the inequality of arithmetic and geometric means. The solution to this simpliest case provides the foundation for finding optimal solutions in all other special cases.

Theorem 1 *If D is the identity matrix, the optimal tiling is:*

$$H \quad = \quad \text{diag}\left(\sqrt[n]{\frac{1}{\nu}}, \cdots, \sqrt[n]{\frac{1}{\nu}}\right) \tag{5}$$

which has the smallest communication volume $\mathcal{V}_{\text{comm}}(H) = n\nu\sqrt[n]{\frac{1}{\nu}}$.

Proof. This is a good example to use Dijkstra's proof style [6].

$$\begin{array}{ll}
 & (\frac{1}{\nu})^{\frac{1}{n}} \\
= & \{ \text{ Constraint } \frac{1}{|\det(H)|} = \nu \ \} \\
 & |\det(H)|^{\frac{1}{n}} \\
\leqslant & \{ \text{ The Hadamard inequality } \} \\
 & (\| h_1 \| \times \cdots \times \| h_n \|)^{\frac{1}{n}} \\
= & \{ \text{ Definition of the Euclidean norm; } H \geqslant 0 \} \\
 & \left(\sqrt{h_{1,1}^2 + \cdots + h_{1,n}^2} \times \cdots \times \sqrt{h_{n,1}^2 + \cdots + h_{n,n}^2}\ \right)^{\frac{1}{n}} \\
\leqslant & \{ \text{For nonnegative } x_1, \cdots, x_n,\ \sqrt{x_1^2 + \cdots + x_n^2} \leqslant (x_1 + \cdots + x_n) \ \} \\
 & ((h_{1,1} + \cdots + h_{1,n}) \times \cdots \times (h_{n,1} + \cdots + h_{n,n}))^{\frac{1}{n}} \\
\leqslant & \{ \text{ Lemma 1 } \} \\
 & \frac{(h_{1,1}+\cdots+h_{1,n})+\cdots+(h_{n,1}+\cdots+h_{n,n})}{n} \\
= & \{ \sum \} \\
 & \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} h_{i,j}}{n} \\
= & \{ \mathcal{F} \} \\
 & \frac{\mathcal{F}(H)}{n}
\end{array}$$

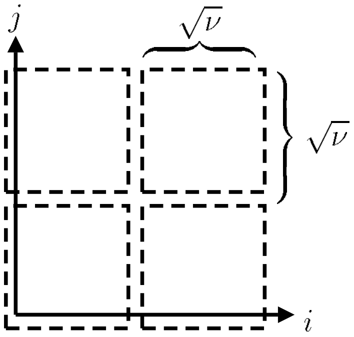


Figure 4: Optimal tiling when $D \in \mathbb{Z}^{n \times n}$ is the identity matrix ($n = 2$).

In the Hadamard inequality, the sign of equality holds if and only if $h_1, \cdots, h_n$ are an orthogonal basis. Since $H \geqslant 0$, the rows of H are mutually orthogonal if and only if H is a diagonal matrix (up to row permutations):

$$H \ = \ \text{diag}(h_{1,1}, \cdots, h_{n,n})$$

If H is diagonal, the second "$\leqslant$" in the above proof steps can be replaced with "=". So,

$$(\frac{1}{\nu})^{\frac{1}{n}} \ = \ |\det(H)|^{\frac{1}{n}} \ = \ (h_{1,1} \times \cdots \times h_{n,n})^{\frac{1}{n}} \ \leqslant \ \frac{h_{1,1} + \cdots + h_{n,n}}{n} \ = \frac{\mathcal{F}(H)}{n}$$

By Lemma 1, the sign of equality holds if and only if $h_{1,1} = \cdots = h_{n,n} = \sqrt[n]{\frac{1}{\nu}}$. This means that (5) is the optimum, and it yields the minimal communication volume $\mathcal{V}_{\text{comm}}(H) = nv\sqrt[n]{\frac{1}{\nu}}$. ■

In this special case, both the dependence cone $\mathcal{C}(D)$ and the tiling cone $\mathcal{C}(h)$ are the first orthant in the Euclidean space, a special form of a pointed cone. Note that $P = H^{-1} = \text{diag}(\sqrt[n]{\nu}, \cdots, \sqrt[n]{\nu})$. In the optimal tiling, the iteration space is tiled with rectangles of size $\sqrt[n]{\nu} \times \cdots \times \sqrt[n]{\nu}$ with the edges of a tile parallel to the natural axes, i.e., the edges of the first orthant cone. This is illustrated in Figure 4 for a two-dimensional iteration space.

5.3 D Is a Full Rank Square Matrix

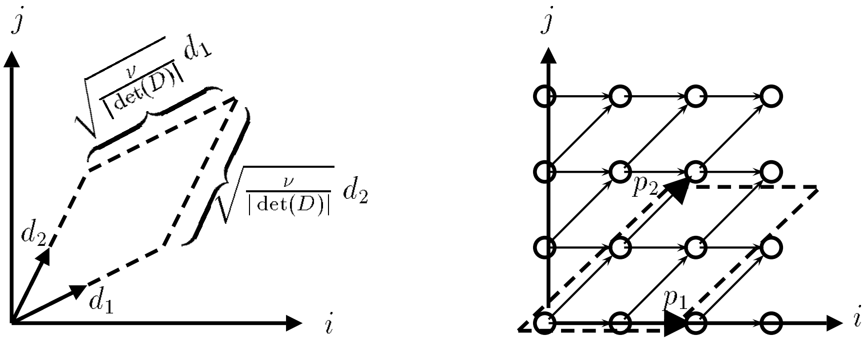
Theorem 2 *If D is a full rank square matrix, the optimal tiling is:*

$$H \ = \ \sqrt[n]{\frac{|\det(D)|}{\nu}}\, D^{-1} \tag{6}$$

which has the smallest communication volume $\mathcal{V}_{\text{comm}}(H) = n\nu\sqrt[n]{\frac{|\det(D)|}{\nu}}$.

Proof. If D is a full rank square matrix, so is HD. If we let $HD = H'$, which implies $\det(H)\det(D) = \det(H')$, we can reduce the optimisation problem (3) to:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H') = \sum_{i=1}^{n}\sum_{j=1}^{m} h'_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H')|} = \frac{\nu}{|\det(D)|} \\ & H' \geqslant 0 \end{array}$$



(a) For $D = [d_1, d_2]$ (b) For Example 4

Figure 5: Optimal tiling when $D \in \mathbb{Z}^{n\times n}$ is a full rank square matrix ($n = 2$).

We are back to the case we solved before when the dependence matrix is the identity matrix. By Theorem 1, the optimal solution to this problem is:

$$H' \;=\; \text{diag}\left(\sqrt[n]{\frac{|\det(D)|}{\nu}}, \cdots, \sqrt[n]{\frac{|\det(D)|}{\nu}}\right)$$

Since $HD = H'$, (6) is the optimum to (3) and it attains the smallest communication volume $\mathcal{V}_{\text{comm}}(H) = n\nu\sqrt[n]{\frac{|\det(D)|}{\nu}}$. ∎

Geometrically, this special case can be reduced to the previous case when D is the identity by a change of basis. In the column basis of D, D becomes the identity matrix.

In this special case, we have

$$P \;=\; H^{-1} \;=\; \sqrt[n]{\frac{\nu}{|\det(D)|}}\, D$$

The dependence cone has n edges with the columns of D as the n edges. Also, the tiling cone has n edges with the rows of D^{-1} as the n edges. In the optimal tiling, the iteration space is tiled with parallelepipeds of size $\sqrt[n]{\frac{\nu}{|\det(D)|}}\, d_1 \times \cdots \times \sqrt[n]{\frac{\nu}{|\det(D)|}}\, d_n$ with its edges parallel to the columns of D. Figure 5(a) illustrates the optimal tiling for a two-dimensional iteration space by depicting the shape and size of the tile at the origin.

Example 4 By Theorem 2, the optimal tiling for the dependence matrix in Example 1 is:

$$H \;=\; \sqrt{\frac{1}{\nu}}\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \qquad P \;=\; \sqrt{\nu}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

This is illustrated in Figure 5(b) when $\nu = 4$.

Note that diagonal matrices are a special form of square matrices. Thus, Theorem 2 also delivers the optimal tiling in the case when the dependence matrix D is diagonal.

5.4 D Is Nonnegative and Contains a Diagonal Matrix

This section considers the case when D is nonnegative and contains a square submatrix as a diagonal submatrix. Without loss of generality, we assume that D is of the form $D = [D_1, D_2]$ such that $D_1 \in \mathbb{Z}^{n\times n}$ is diagonal. Let

$$\tilde{D} \;=\; \text{diag}(\sum_{j=1}^{m} d_{1,j}, \cdots, \sum_{j=1}^{m} d_{n,j})$$

That is, $\tilde{D}$ is the diagonal matrix whose i-th diagonal element is the sum of the i-th row of D. Note that $\tilde{D} > 0$ since $D_1 > 0$.

Theorem 3 *If $D = [D_1, D_2]$ as defined above, the optimal tiling is:*

$$H \;=\; \sqrt[n]{\frac{\det(\tilde{D})}{\nu}}\,\tilde{D}^{-1} \tag{7}$$

which has the smallest communication volume $\mathcal{V}_{\text{comm}}(H) = n\nu\sqrt[n]{\frac{\det(\tilde{D})}{\nu}}$.

Proof. When $D = [D_1, D_2]$, the optimisation problem (3) can be rewritten as follows:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n}\sum_{j=1}^{m}(H[D_1, D_2])_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & H[D_1, D_2] \geqslant 0 \end{array}$$

$H[D_1, D_2] \geqslant 0$ can be decomposed into $HD_1 \geqslant 0$ and $HD_2 \geqslant 0$. Since $D_1 > 0$ is diagonal, $HD_1 \geqslant 0$ implies $H \geqslant 0$. Since D_2 is nonnegative, $H \geqslant 0$ implies $HD_2 \geqslant 0$. By noting further that $\sum_{i=1}^{n}\sum_{j=1}^{m}(H[D_1, D_2])_{i,j} = \sum_{i=1}^{n}\sum_{j=1}^{n}(H\tilde{D})_{i,j}$, we can simplify the above problem to:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n}\sum_{j=1}^{n}(H\tilde{D})_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & H \geqslant 0 \end{array}$$

which is equivalent to:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n}\sum_{j=1}^{n}(H\tilde{D})_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & H\tilde{D} \geqslant 0 \end{array}$$

because both $H \geqslant 0$ and $H\tilde{D} \geqslant 0$ are equivalent when $\tilde{D} > 0$ is diagonal, We are back to the case we solved before when the dependence matrix is a square matrix. By Theorem 2, the tiling given in (7) is optimal and has the smallest communication volume as indicated. ∎

In this special case, we have

$$P \;=\; H^{-1} \;=\; \sqrt[n]{\frac{\nu}{\det(\tilde{D})}}\,\tilde{D}$$

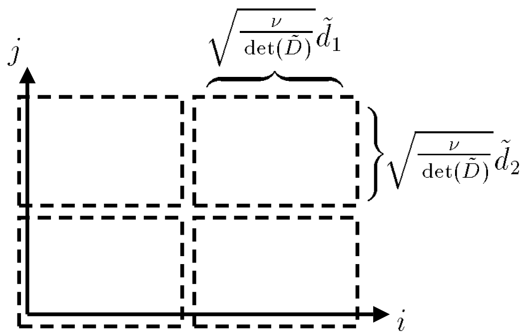


Figure 6: Optimal tiling when $D \in \mathbb{Z}^{n\times m}$ is nonnegative and contains a diagonal matrix ($n = 2$).

When $D = [D_1, D_2]$, the dependence cone and tiling cone are the first orthant. The optimal tiling consists of dividing the iteration space using rectangles of size $\sqrt[n]{\frac{\nu}{\det(\tilde{D})}}\ \tilde{d}_1 \times \cdots \sqrt[n]{\frac{\nu}{\det(\tilde{D})}}\ \tilde{d}_n$, with the edges of a tile parallel to the natural axes. The dependence vectors in D_1 completely determines the shape of a tile (the rectangular tiling), while those in D_1 contribute only to determining the aspect ratios of a tile – the larger the sum of the i-th entries of all dependence vectors, the longer of the side of the tile along the i-th dimension. Figure 6 illustrates the optimal tiling for a two-dimensional iteration space by depicting the shape and size of the tile at the origin.

Example 5 Consider a double loop with the dependence matrix:

$$D = [D_1, D_2] = \begin{bmatrix} 2 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

where D_1 is diagonal. We find that

$$\tilde{D} = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}$$

An application of Theorem 3 yields the optimal tiling:

$$H = \begin{bmatrix} \sqrt{\frac{3}{5\nu}} & 0 \\ 0 & \sqrt{\frac{5}{3\nu}} \end{bmatrix} \qquad P = \begin{bmatrix} \sqrt{\frac{5\nu}{3}} & 0 \\ 0 & \sqrt{\frac{3\nu}{5}} \end{bmatrix}$$

5.5 D Contains All n Extremal Rays of the Dependence Cone

In this special case, the dependence cone $\mathcal{C}(D)$ has exactly n extremal rays and the columns of D can always be permuted so that $D = [D_1, D_2]$, where $D_1 \in \mathbb{Z}^{n\times n}$ is nonsingular and its columns are the n extremal rays of the dependence cone. Using the notations in (4), we have $D' = D_1$ and $T = (D_1^{-1})^T$.

By the definition of D_1, $D_1^{-1}[D_1, D_2] = [\mathcal{I}, D_1^{-1}D_2]$ is nonnegative and contains the identity as a submatrix. This will enable us to reduce this case to the one solved in Section 5.4.

Example 6 Consider a triple loop with the dependence matrix:

$$D = \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

The last column is a positive linear combination of the first three: $d_4 = \frac{1}{2}(d_1 + d_2 + d_3)$. So we have $D_1 = [d_1, d_2, d_3]$ and $D_2 = [d_4]$. Thus, the dependence cone, as shown in Figure 7, has three edges d_1, d_2 and d_3, and three faces are identified by their normals $d_2 \times d_3$, $d_1 \times d_2$ and $d_1 \times d_3$.

This special case is identified for two reasons. First, it provides insights into finding optimal tilings in the general case, to be described in Section 5.6. Second, the closed-form optimal tiling for a 2-deep loop nest can be found more efficiently than if the approach for the general case is

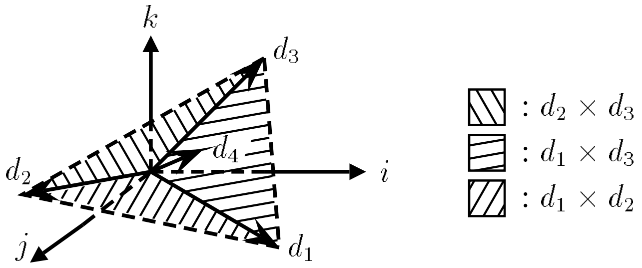


Figure 7: The dependence cone for Example 6.

used. This is because if $D \in \mathbb{Z}^{2\times m}$, we can find D_1 in $O(m)$ time, where m is the number of dependence vectors in D. In fact, the two columns in D_1 can be chosen as the vectors with the largest and smallest ratios $\frac{a}{b}$ among all columns $\begin{bmatrix} a \\ b \end{bmatrix}$ in D.

Theorem 4 *Let $D = [D_1, D_2]$ as defined above. Let $\tilde{D}$ be the diagonal matrix whose i-th diagonal element is the sum of the i-th row of $[\mathcal{I}, D_1^{-1}D_2]$. Then the optimal tiling is:*

$$H \quad = \quad \sqrt[n]{\frac{\det(\tilde{D})|\det(D_1)|}{\nu}}\,\tilde{D}^{-1}D_1^{-1} \tag{8}$$

which has the smallest communication volume $\mathcal{V}_{\text{comm}}(H) = n\nu\sqrt[n]{\frac{\det(\tilde{D})|\det(D_1)|}{\nu}}$.

Proof. When $D = [D_1, D_2]$, the optimisation problem (3) to be solved is as follows:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H) = \sum_{i=1}^{n}\sum_{j=1}^{m}(H[D_1, D_2])_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H)|} = \nu \\ & H[D_1, D_2] \geqslant 0 \end{array}$$

Letting $HD_1 = H'$, we have $H[D_1, D_2] = HD_1D_1^{-1}[D_1, D_2] = HD_1[\mathcal{I}, D_1^{-1}D_2] = H'[\mathcal{I}, D_1^{-1}D_2]$. So we can reduce the above problem to:

$$\begin{array}{rl} \textbf{Minimise} & \mathcal{F}(H') = \sum_{i=1}^{n}\sum_{j=1}^{m}(H'[\mathcal{I}, D_1^{-1}D_2])_{i,j} \\ \textbf{Subject to} & \frac{1}{|\det(H')|} = \frac{\nu}{|\det(D_1)|} \\ & H'[\mathcal{I}, D_1^{-1}D_2] \geqslant 0 \end{array}$$

Because D_1 are the extremal rays of the dependence cone, $D_1^{-1}D_2$ must be a nonnegative matrix. Thus, we are back to the previous case we solved before when the dependence matrix is nonnegative and contains a diagonal matrix. By Theorem 3, the optimal solution for H' is:

$$H' \quad = \quad \sqrt[n]{\frac{\det(\tilde{D})|\det(D_1)|}{\nu}}\,\tilde{D}^{-1}$$

A further use of the fact $HD_1 = H'$ concludes the proof of this theorem. ∎

Example 7 Continuing the example in Example 6, we find that

$$[\mathcal{I}, D_1^{-1}D_2] = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right] \qquad \tilde{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

We use Theorem 4 to derive the following optimal tiling:

$$H = \frac{1}{\sqrt[3]{4\nu}} \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \qquad P = \sqrt[3]{4\nu} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

In this special case, we have

$$P = H^{-1} = \sqrt[n]{\frac{\nu}{\det(\tilde{D})|\det(D_1)|}}\, D_1\tilde{D}$$

In the optimal tiling, the iteration space is tiled with parallelepipeds whose n edges are parallel to the n extremal rays of the dependence cone. In more detail, the dependence vectors in D_1 completely determine the shape of a tile, and the remaining dependence vectors (i.e., those in D_2) have effects only on the aspect ratios of a tile.

Two final remarks about how to select D_1 in Theorem 4 are in order.

(a) D_1 is unique up to multiplication by positive scalars. Assume that D contains two different submatrices $A \in \mathbb{Z}^{n\times n}$ and $B \in \mathbb{Z}^{n\times n}$ such that the columns of each submatrix are n linearly independent extremal rays for the dependence cone. We must have $AQ_1 = B$ and $BQ_2 = A$, where Q_1 and Q_2 are nonsingular nonnegative square matrices. This implies that $Q_1 = Q_2^{-1}$. Being both nonnegative, Q_1 and Q_2 must be both diagonal. Hence, D_1 is unique up to multiplication by positive scalars.

(b) Continuing from (a), A and B must be such that $A = sB$, where s is a positive scaling factor. It is not difficult to see from (8) that the same optimal tiling H is obtained no matter whether A or B is used as D_1 in Theorem 4.

5.6 D Has Full Row Rank

The four special cases discussed above share the following properties in common:

- The dependence cone has n rays and is generated by n columns of D. According to Section 5.1, both the tiling and dependence cone have exactly n rays (edges) and n faces.

- The optimal tiling is unique in each case.

- In the optimal tiling, the n edges of a tile are parallel to the n edges of the dependence cone.

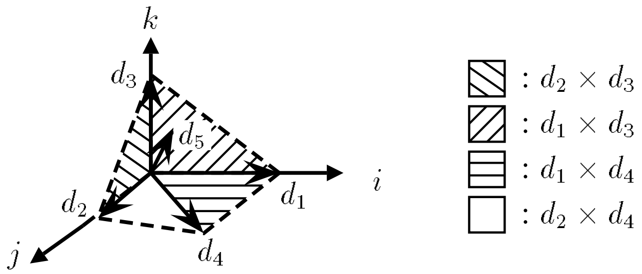


Figure 8: The dependence cone for Example 8 (with 4 edges and 4 faces).

- The optimal tiling has a closed-form expression.

All these properties may not hold in the case when the dependence cone has more than n extremal rays (or edges). In this case, the dependence cone cannot be generated by using only n columns of D. As a result, several and sometimes infinitely many optimal tilings may exist.

Example 8 Consider the dependence matrix:

$$D = \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \left[\begin{array}{ccccc} 2 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{array}\right] \end{array}$$

The dependence cone, shown in Figure 8, has four extremal rays (edges), which are the first four columns of D. Note that $d_5 = \frac{1}{2}d_1 + d_2 + d_3$. For this example, the dependence cone $\mathcal{C}(D)$ cannot be generated by any three dependence vectors in D. In other words, D does not contain three columns that generate the dependence cone. So Theorem 3 cannot be used here.

This section discusses how to find optimal tilings in the general case. The problem was solved in [3]. But, our solution is developed in a different and conceptually simpler framework based on Lemma 1. In addition, several new results are provided concerning the necessity for a tiling to be optimal and the possibility for a program to have infinitely many optimal tilings. All these together provide new insights into the problem of tiling nested loops in general.

As an important result, Lemma 2 shows that a tiling that is optimal must induce the same amount of communication on all faces of a tile. Lemma 3 reduces the problem of finding an optimal tiling to one of finding a matrix with the largest determinant (in absolute value). As a new result, Lemma 4 provides a necessary condition for a tiling to be optimal, showing that a tiling has the largest determinant (in absolute value) only if all its row vectors are contained in the faces of the tiling cone. Lemma 5 assures us the existence of an optimal tiling with its rows being all extremal rays of the tiling cone. This restricts the search space for optimal tilings to a finite set of matrices whose rows are all extremal rays in the tiling cone. These optimal tilings are called the *extremal-ray optimal tilings*. There can be other optimal tilings as well, which will be infinitely many whenever they exist. This new result is given in Lemma 6, which provides a necessary and sufficient condition for a program to have infinitely many optimal tilings. Finally, these results are summarised in Theorem 5. This section also provides a geometric interpretation behind an extremal-ray optimal tiling. Section 6 describes an efficient procedure for generating all extremal-ray optimal tilings.

Let h be a tiling vector in the tiling cone $\mathcal{C}(h)$. We define:

$$c(h) \;=\; \sum_{j=1}^{m}(hD)_j$$

If h_i is the i-th row of a tiling matrix H, $\frac{c(h_i)}{|\det(H)|}$ represents the communication volume going through the face h_i.

The following lemma is proved using the inequality of arithmetic and geometric means.

Lemma 2 *Assume that D has full row rank. If H is an optimal tiling, then all faces of a tile sustain the same amount of communication, i.e., $c(h_1) = \cdots = c(h_n)$.*

Proof. We construct a tiling H' from H as follows:

$$H' \;=\; \text{diag}\left(\frac{\sqrt[n]{c(h_1)\times\cdots\times c(h_n)}}{c(h_1)},\cdots,\frac{\sqrt[n]{c(h_1)\times\cdots\times c(h_n)}}{c(h_n)}\right) H$$

It is clear that $|\det(H')| = |\det(H)| = \frac{1}{\nu}$ and $HD \geqslant 0$ implies $H'D \geqslant 0$. H and H' yield the following communication volumes for a tile, respectively:

$$\begin{aligned}\mathcal{V}_{\text{comm}}(H) &= \nu(c(h_1)+\cdots+c(h_n))\\ \mathcal{V}_{\text{comm}}(H') &= n\nu\sqrt[n]{c(h_1)\times\cdots\times c(h_n)}\end{aligned}$$

By Lemma 1, $\mathcal{V}_{\text{comm}}(H') \leqslant \mathcal{V}_{\text{comm}}(H)$ and the sign of equality holds if and only if $c(h_1) = \cdots = c(h_n)$. This means that if $c(h_1) = \cdots = c(h_n)$ does not hold, then H is not optimal. ■

Let G be the set of all tiling matrices, which are up to row permutations and multiplications by positive scalars, such that each tiling matrix induces the same amount of communication on all faces of a tile. G can be constructed as follows:

$$G \;=\; \{\begin{bmatrix}\frac{h_1}{c(h_1)}\\ \vdots\\ \frac{h_n}{c(h_n)}\end{bmatrix} \mid h_1,\cdots,h_n \in \mathcal{C}(h) \text{ and are linearly independent}\} \qquad (9)$$

If two tiling matrices are such that one is a row permutation of the other, both represent exactly the same tiling to the iteration space. So we include only one of the two in G. If two tiling matrices B_1

and B_2 are identical up to scaling, it is again only necessary to include one of the two in G. This is because that, as will be shown in Lemma 3 below, an optimal tiling must have the form of (10), implying that the same H in (10) results regardless of if B_1 or B_2 is used as B in (10).

Next, the problem of finding an optimal tiling is reduced to one of finding a matrix in G with the largest determinant in absolute value.

Lemma 3 *Assume that D has full row rank. An optimal tiling*

$$H = \sqrt[n]{\frac{1}{\nu|\det(B)|}}\,B, \quad \text{where } B \in G \tag{10}$$

has the largest $|\det(B)|$, yielding the smallest communication volume $\mathcal{V}_{\mathrm{comm}}(H) = n\nu\sqrt[n]{\frac{1}{\nu|\det(B)|}}$.

Proof. By Lemma 2 and by definition of G, all optimal tilings are contained in the set:

$$\{\sqrt[n]{\frac{1}{\nu|\det(B)|}}\,B \mid B \in G\}$$

Note that the communication volume of a tile induced by H is $\mathcal{V}_{\mathrm{comm}}(H) = n\nu\sqrt[n]{\frac{1}{\nu|\det(B)|}}$. Hence, H is optimal if and only if it has the largest $|\det(B)|$. ■

The following lemma provides a necessary condition for a tiling to be optimal, i.e., for a tiling to have the largest determinant (in absolute value).

Lemma 4 *If $B \in G$ is a tiling matrix with at least one row vector not contained in the faces of the tiling cone, then B is not optimal, i.e., B does not have the largest $|\det(B)|$.*

Proof. If B contains a row vector that is not contained in the faces of the tiling cone, it is always possible to find n linearly independent rays $r_1, \cdots, r_n$ in the tiling cone such that the row vectors $b_1, \cdots, b_n$ of B are contained in cone$(r_1, \cdots, r_n)$ and cone$(b_1, \cdots, b_n)$ is strictly contained in cone$(r_1, \cdots, r_n)$. Let B' be the matrix formed with its i-th row $b'_i = \frac{r_i}{c(r_i)}$. By construction, $c(b'_1) = \cdots = c(b'_n) = 1$. Since cone$(b_1, \cdots, b_n) \subset$ cone$(r_1, \cdots, r_n) =$ cone$(b'_1, \cdots, b'_n)$, there must exist an $n \times n$ nonsingular nonnegative matrix Q such that $B = QB'$. The rest of the proof is to show that $|\det(Q)| < 1$, i.e., $|\det(B)| < |\det(B')|$, implying that B is not optimal by Lemma 3. From the construction of B, we have $c(b_1) = \cdots = c(b_n) = 1$. Since $B = QB'$, an algebraic manipulation shows that $c(b_i) = \sum_{j=1}^{n} q_{i,j}c(b'_j)$. Thus, $q_{i,1} + \cdots + q_{i,n} = 1$. Hence, all entries $q_{i,j}$ of Q must be $0 \leqslant q_{i,j} \leqslant 1$. For the i-th row q_i of Q, we have $\| q_i \| = \sqrt{q_{i,1}^2 + \cdots + q_{i,n}^2} \leqslant q_{i,1} + \cdots + q_{i,n} = 1$. By the Hadamard inequality, $|\det(Q)| \leqslant \| q_1 \| \times \cdots \times \| q_n \| \leqslant 1$. The sign of equality in $|\det(Q)| \leqslant 1$ holds if and only if Q is the identity matrix up to permutations in order for $q_1, \cdots, q_n$ to be mutually orthogonal. But this will imply that the rows of $B = QB'$ are all contained in the faces of the tiling cone, contradicting the given assumption. Hence, $|\det(Q)| < 1$, implying $|\det(B)| = |\det(Q)||\det(B')| < |\det(B')|$. That is, B is not optimal by Lemma 3. ■

Based on Lemma 4 alone, there are still infinitely many matrices in G to be examined for optimal tilings. The following lemma shows the existence of an optimal tiling with its rows being all extremal rays of the tiling cone. This allows us to restrict the search space for optimal tilings to be a finite set of matrices whose rows are all extremal rays in the tiling cone.

Lemma 5 *Let $B \in G$ be an optimal tiling. Then, there exists an optimal tiling $B' \in G$ such that the rows of B' are all extremal rays in the tiling cone and such that $\det(B) = \det(B')$.*

Proof. Let X be the $p \times n$ matrix such that its i-row $x_i = \frac{r_i}{c(r_i)}$, where $r_1, \cdots, r_p$ are all p different extremal rays in the tiling cone. By construction, $c(x_1) = \cdots = c(x_p) = 1$. Since $B \in G$, there must exist an $n \times p$ nonnegative matrix Q such that $B = QX$. That is, the i-th row b_i of B is a nonnegative linear combination of the extremal rays of the tiling cone: $b_i = \sum_{j=1}^{p} q_{i,j} x_j$, where $q_i = (q_{i,1}, \cdots, q_{i,p})$ is the i-th row of Q. An algebraic calculation shows that $c(b_i) = \sum_{j=1}^{p} q_{i,j} c(x_j)$. From the construction of B, we have $c(b_1) = \cdots = c(b_n) = 1$. By further using the fact that $c(x_1) = \cdots = c(x_p) = 1$, we obtain $\sum_{j=1}^{p} q_{i,j} = 1$. In the rest of the proof, we show how to derive from B an optimal tiling B' with the property stated in the lemma. Assume that q_i contains more than one non-zero entries, in which case, b_i is not an extremal ray. We let $B' = Q'X$, where Q' is the matrix with all its rows taken from Q except that the entries in the i-th row $q'_i = (q'_{i,1}, \cdots, q'_{i,p})$ are treated as variables. We consider the following linear programming problem:

$$\begin{array}{rl} \textbf{Maximise} & \det(B') = \det(Q'X) \\ \textbf{Subject to} & \sum q'_{i,j} = 1 \\ & q'_{i,1} \geqslant 0, \cdots, q'_{i,p} \geqslant 0 \end{array} \tag{11}$$

We are given that $B = QX$ is an optimal tiling. Thus, $q'_i = q_i$ must be an optimal solution to (11). But the optimum can also be attained at one of the p vertices of the solution space: $(1, 0, \cdots, 0), (0, 1, 0, \cdots, 0), \cdots, (0, \cdots, 0, 1)$. Let q'_i be such a vertex. Then $B' = Q'X$ is also an optimal tiling such that its i-th row is an extremal ray of the tiling cone and such that $\det(B) = \det(B')$. By repeating the process for a total of at most n times, we will obtain an optimal tiling B' with the property stated in the lemma. ■

There can be optimal tilings other than the extremal-ray optimal tilings. Whenever that happens, there will be infinitely many optimal tilings, and vice versa.

Lemma 6 *There are infinitely many optimal tilings if and only if there exists an optimal tiling $B \in G$ such that not all its rows are extremal rays of the tiling cone.*

Proof. The "only if" part is simple. The set of tiling matrices whose rows are all extremal rays of the tiling cone is finite, because the number of the rays is finite. To prove the "if" part, let us assume that the i-th row of B is a nonnegative linear combination of at least two extremal rays. Then, by proceeding exactly as we did in the proof of Lemma 5, we can find an optimal tiling B' that differs from B only in their i-th rows and satisfies $\det(B) = \det(B')$. Then, the set $\{\alpha B + (1-\alpha)B' \mid 0 \leqslant \alpha \leqslant 1\}$ contains infinitely many optimal tilings. ■

Let R be the set of all extremal rays in the tiling cone $\mathcal{C}(h)$. Let G' be the subset of G in (9) and be defined as follows:

$$G' = \{ \begin{bmatrix} \frac{r_1}{c(r_1)} \\ \vdots \\ \frac{r_n}{c(r_n)} \end{bmatrix} \mid r_1, \cdots, r_n \in R\} \tag{12}$$

R contains a finite number of rays. So G' contains a finite number of matrices, given by $\binom{|R|}{n}$. According to (4), every tiling matrix H in G' satisfies the atomic tiles constraint $HD \geqslant 0$.

Theorem 5 *Assume that D has full row rank. There exists an optimal tiling of the form:*

$$H = \sqrt[n]{\frac{1}{\nu|\det(B)|}}\,B, \quad \text{where } B \in G'$$

Proof. Lemmas 2, 3 and 5. ■

Example 9 Continuing Example 8, we find the four extremal rays in the tiling cone $\mathcal{C}(h)$:

$$r_1 = (0,1,0), \quad r_2 = (0,2,1), \quad r_3 = (1,0,0), \quad r_4 = (2,0,1)$$

from which we construct the four matrices contained in G' (up to row permutations):

$$B_1 = \begin{bmatrix} 0 & 1/3 & 0 \\ 0 & 2/6 & 1/6 \\ 1/4 & 0 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 1/3 & 0 \\ 0 & 2/6 & 1/6 \\ 2/8 & 0 & 1/8 \end{bmatrix}$$
$$B_3 = \begin{bmatrix} 0 & 1/3 & 0 \\ 1/4 & 0 & 0 \\ 2/8 & 0 & 1/8 \end{bmatrix} \quad B_4 = \begin{bmatrix} 0 & 2/6 & 1/6 \\ 1/4 & 0 & 0 \\ 2/8 & 0 & 1/8 \end{bmatrix}$$

We find that $|\det(B_1)| = |\det(B_2)| = 1/72$ and $|\det(B_3)| = |\det(B_4)| = 1/96$. By Theorem 5, there are two extremal-ray optimal tilings:

$$H_1 = \sqrt[3]{\frac{72}{\nu}}B_1 \quad H_2 = \sqrt[3]{\frac{72}{\nu}}B_2$$

From the proof of Lemma 6, all tilings in the following (infinite) set are also optimal:

$$\{\sqrt[3]{\frac{72}{\nu}}(\alpha B_1 + (1-\alpha)B_2) \mid 0 \leqslant \alpha \leqslant 1\}$$

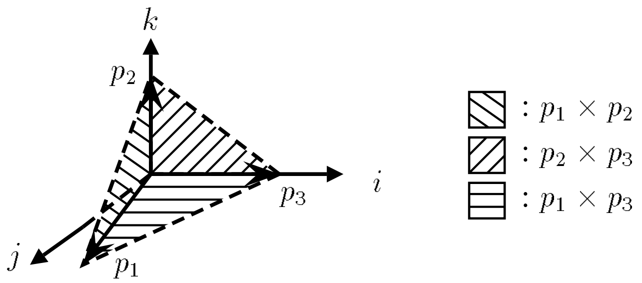


Figure 9: The cone generated by the columns of $P = [p_1, p_2, p_3] = H_1^{-1}$ for Example 9. This cone contains the dependence cone in Figure 8 with the two faces of the same filling style coincide.

Let us now provide a geometric intuition behind an extremal-ray optimal tiling when the dependence cone has more than n edges. Recall our discussions in Section 5.1 that if the dependence cone has more than n edges, it must also have more than n faces, and vice versa.

In an optimal tiling

$$H = \sqrt[n]{\frac{1}{\nu|\det(B)|}}\, B \quad P = \sqrt[n]{\nu|\det(B)|}\, B^{-1}$$

the columns of H^{-1} generates a cone such that B has the largest $|\det(B)|$. This cone contains the dependence cone with its n faces coincident with some n faces of the dependence cone. The iteration space is tiled with parallelepipeds whose n edges are parallel to the n edges of this cone. The shape of a tile is completely determined by the dependence vectors of D that define B and the other dependence vectors have effects only on the aspect ratios of a tile. Take the optimal tiling H_1 for example:

$$H_1^{-1} = \sqrt[3]{\frac{\nu}{72}} \begin{bmatrix} 0 & 0 & 4 \\ 3 & 0 & 0 \\ -6 & 6 & 0 \end{bmatrix}$$

Figure 9 depicts the cone generated by the columns of H_1^{-1} and illustrates its relation with the dependence cone in Figure 8.

In the general case, it is also possible to have just one unique optimal tiling. For the dependence matrix in Example 3, the optimal tiling is $H = \sqrt[4]{\frac{1}{\nu|\det(B)|}} B$, where $B = \frac{1}{c(r_1)\times\cdots\times c(r_4)}[r_1, \cdots .r_4]^T$.

6 A Procedure for Finding Extremal-Ray Optimal Tilings

By putting all results in the paper together, Figure 10 gives a procedure for finding extremal-ray optimal tilings. The procedure consists of a sequence of **if** statements: The first four **if** statements construct the closed-form optimal tilings for the four special cases discussed in the paper. The **else** part captures the general case when the dependence matrix D has full row rank and is described below.

6.1 The else Part

Step (a) constructs the extremal rays for the tiling cone $\mathcal{C}(h)$. As mentioned in Section 5.1, these rays can be constructed from the faces of the dependence cone $\mathcal{C}(D)$. Every set of $n-1$ linearly independent columns $x_1, \cdots, x_{n-1}$ in D potentially defines a face for the dependence cone. Let r be the normal of the hyperplane spanned by the $n-1$ columns: $r = x_1 \times \cdots \times x_{n-1}$ (up to scaling). If $rD \geqslant 0$, then r is a ray. We must also check to see if $rD \leqslant 0$ because that is the case, then $-r$ is a ray. Otherwise, the $n-1$ columns do not define a face for the dependence cone and are ignored.

We propose to use the (row) echelon reduction to construct the extremal rays for the tiling cone. Let z_i be the column containing the first nonzero in row i, where $z_i = \infty$ if the row is entirely zero. A matrix is in *row echelon form* if (a) $z_{i+1} > z_i$ if row i is not entirely zero, and (b) row $i+1$ is entirely zero if row i is. In line 19, we set up A as the $n \times (n-1)$ matrix whose columns are $n-1$ columns $x_1, \cdots, x_{n-1}$ of D. In line 20, we reduce A to row echelon form, which consists of finding a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ and a row echelon matrix $S \in \mathbb{Z}^{n \times (n-1)}$ such that $UA = S$.

The idea of using the echelon reduction to obtain the rays has three advantages:

- We can detect linear independence of the columns in $A = [x_1, \cdots, x_{n-1}]$, and if that is the case, obtain the normal r to the hyperplace spanned by $x_1, \cdots, x_{n-1}$ at the same time. Indeed, $x_1, \cdots, x_{n-1}$ are linearly independent if and only if S contains the last row as its only row that is entirely zero, in which case $r = u_n$ is the last row of U and r can be then checked in lines 23 – 27 to see if it is a ray or not.

- We can easily detect and remove redundant rays constructed from different $n \times (n-1)$ submatrices of D in the **for** loop of Step (a). Let A_1 and A_2 be two $n \times (n-1)$ matrices each contains $n-1$ linearly independent columns in D. When reducing A_1 and A_2 to row echelon forms S_1 and S_2, we will find two unimodular matrices U and V such that $UA_1 = S_1$ and $VA_2 = S_2$. If both A_1 and A_2 have different column spaces, u_n and v_n are not co-linear. If both A_1 and A_2 have the same column space, we must have $u_n = \pm 1 v_n$, where the scaling factor is ± 1. Otherwise U and V cannot be both unimodular. Our implementation exploits this property to suppress all but one in each set of identical extremal rays.

- The echelon reduction is very efficient on matrices with small integers. Hence, it suits our needs extremely well because the entries of the dependence matrix D are small integers with many -1, 0 and 1.

Having obtained the extremal ways, Step (b) applies Theorem 5 to enumerate all optimal tilings. Let p be the number of rays in the tiling cone, i.e., $p = |R|$. This step involves computing

```
procedure OptTilingsGen
input: The dependence matrix $D \in \mathbb{Z}^{n\times m}$ with full row rank ($m \geqslant n$)
output: All extremal-ray optimal tilings $H$ to (3)

if $D$ is the identity matrix then
    $H$ is (5);
else If $D$ is a square matrix then
    $H$ is (6);
else if $D$ is nonnegative and contains a diagonal submatrix $D_1 \in \mathbb{Z}^{n\times n}$ then
    Permute the columns of $D$ so that $D = [D_1, D_2]$;
    $H$ is (7);
else if $D \in \mathbb{Z}^{2\times m}$ then
    Permute the columns of $D$ so that $D=[D_1, D_2]$, where the columns of $D_1 \in \mathbb{Z}^{2\times 2}$
    are the vectors with the largest and smallest ratios $\frac{a}{b}$ among all $\begin{bmatrix} a \\ b \end{bmatrix}$ in $D$;
    $H$ is (8);
else      /* $D$ has full row rank */
    (a) Construct the set $R$ of the extremal rays for the tiling cone $\mathcal{C}(h)$;
        $R := \emptyset$;
        for every set of $n-1$ columns $x_1, \cdots, x_{n-1}$ in $D$ do
            Let $A = [x_1, \cdots, x_{n-1}]$ be an $n \times (n-1)$ matrix;
            Reduce $A$ to row echelon form $S$, i.e, find a unimodular matrix
            $U \in \mathbb{Z}^{n\times n}$ such that $UA = S$, where $S$ is in row echelon form;
            if the last row of $S$ is the only row of $S$ that is entirely zero then
                /* $x_1, \cdots, x_{n-1}$ are linearly independent */
                if $u_n D \geqslant 0$ then          /* $u_n$ is the $n$-th row of $U$ */
                    $R := R + \{u_n\}$;      /* $u_n = x_1 \times \cdots \times x_{n-1}$ up to scaling */
                else if $u_n D \leqslant 0$ then
                    $R := R + \{-u_n\}$;
                endif
            endif
        endfor
    (b) Construct all optimal tilings $H = \sqrt[n]{\frac{1}{\nu|\det(B)|}}\, B$, where $B \in G'$ in (12), such
        that $B$ has the largest determinant $|\det(B)|$;
        for every set of $n$ extremal rays $r_1, \cdots, r_n$ in $R$ (up to row permutations) do
            Let $B = \frac{1}{c(r_1)\times\cdots\times c(r_n)} \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} = \frac{1}{c(r_1)\times\cdots\times c(r_n)} A$    /* Def. of $G'$ in (12) */
            Compute $\det(A)$ in floating-point using Gauss elimination with pivoting;
            Make $\det(A)$ an integer after permutations on the order of roundoff error;
            $|\det(B)| := \frac{|\det(A)|}{c(r_1)\times\cdots\times c(r_n)}$;
            if $B$ is as good as the existing optimal solution(s) found so far then
                Record $B$ as yet another optimal solution;
            else if $B$ is better than the existing optimal solutions then
                Discard all the existing optimal solutions;
                Record $B$ as a better optimal solution;
            endif
        endfor
endif
```

Figure 10: Procedure for finding communication-minimal tilings.

the determinants for $\binom{p}{n}$ square integer matrices whose rows are rays of the tiling cone. Unlike the dependence matrix D, an extremal ray may contain relatively larger integers. For efficiency considerations, our implementation uses Gauss elimination with pivoting in floating-point operations (line 33). The determinants in real values are then converted to integers by perturbations on the order of roundoff error (line 34). In lines 36 – 41, we pick up optimal solutions incrementally, avoiding an expensive sorting process that would have to be used otherwise.

6.2 Time Complexity

The time complexity of our procedure is dominated by the time complexity of the **else** part, which consists of two steps. The **for** loop in Step (a) has $\binom{m}{n-1}$ iterations, and its loop body is dominated by reducing an $n \times (n-1)$ integer matrix to row echelon form. This echelon reduction has the same time complexity as the polynomial algorithm for finding the Hermite normal form of an integer matrix as described in [18, p. 56]. Let p be the number of rays found in Step (a). The search space for optimal tilings in Step (b) contains $\binom{p}{n}$ matrices of size $n \times n$ (up to row permutations). Assuming that it takes $O(n^3)$ to compute the determinant for an $n \times n$ matrix, the time complexity of Step (b) is $O(n^3\binom{p}{n})$.

6.3 Experiments

We implemented the **else** part of the tiling procedure and measured its execution times on generating optimal tilings. Thus, we considered only dependence matrices $D \in \mathbb{Z}^{n\times m}$ such that $m > n$. For practical applications, the depth of a loop nest rarely exceeds 4. Ignoring the special case when $n = 2$ (lines 12 – 14), We restricted our ourselves to 3-deep, 4-deep and 5-deep loop nests.

The complexity of the tiling procedure depends on n, the depth of the loop nest, m, the number of dependence vectors, and p, the number of extremal rays in the tiling cone. It is possible for two dependence matrices D_1 and D_2 of the *same* size $n \times m$ to generate different numbers of rays in their corresponding tiling cones $\{h \mid hD_1 \geqslant 0\}$ and $\{h \mid hD_2 \geqslant 0\}$. To account for p in our experiments, we ran the tiling procedure on 100 randomly generated dependence matrices D of the same size $n \times m$. We restrict the entries of a dependence matrix to be within the range $[-2, 2]$. This is because the dependence vectors in uniform dependence programs usually consist of small integers with the most entries being -1, 0 and 1.

We implemented our tiling procedure in C and compiled it with all compiler options turned off on a Digital DEC Alpha workstation $600^{5/333}$ with a 333MHZ Alpha CPU and 1 Gbyte memory. The performance results given in Figure 11 are interpreted as follows.

Given two dependence matrices of the same size $n \times m$, the times for finding optimal tilings in both cases can differ greatly. In both cases, it can be assumed that Step (a) of the procedure takes about the same amount of time, by performing row echelon reductions on $\binom{m}{n-1}$ matrices with small entries. But the complexity of Step (b) is $O(n^3\binom{p}{n})$, a function of n and p. For a fixed pair of n and m, the larger the number p of rays, the longer Step (b) runs. Therefore, the best (worst) case is reached when the dependence matrix induces the smallest (largest) number of rays for the tiling cone.

For a fixed n, our experimental results seem to indicate that the execution time of the tiling procedure increases as the number m of dependences increases. The reason for this is that for our randomly generated dependence matrices, the number p of rays of the tiling cone tends to increase as m increases. However, it is possible that for two dependence matrices $D_1 \in \mathbb{Z}^{n\times m_1}$ and $D_2 \in \mathbb{Z}^{n\times m_2}$ such that $m_1 > m_2$, the execution time for the former is shorter. This can happen if the tiling cone in the former case has far fewer rays. In one experiment, we used the dependence matrix $D \in \mathbb{Z}^{4\times 7}$ with the best case performance in Figure 11 and constructed from it a new matrix $D' \in \mathbb{Z}^{4\times 14}$ by duplicating its columns. We found that the execution time for processing D' is 9.5 milliseconds, better than the timing results for $D \in \mathbb{Z}^{4\times 11}$ and $D \in \mathbb{Z}^{4\times 12}$. This means that our timing measurements are conservative estimates on the efficiency of the tiling procedure.

With the remarks above in mind, Figure 12 depicts in curves the average performance of the tiling procedure by using the data from Figure 11. As can be seen from the performance figures, the tiling procedure is extremely efficient for uniform dependence programs that arise in practical applications. In our implementation, we made no attempts in optimising the code either by hand or using optimising options from the compiler. For academic purposes, it is possible to improve the performance of the tiling procedure on more than 4-deep loop nests by a good engineering and/or using a suboptimal algorithm to implement Step (b) of the tiling procedure. One heuristics-based solution is the subset selection algorithm described in [10] and suggested in [17].

6.4 Making H^{-1} Integral

A tiling transformation H must be constructed such that H^{-1} is integral. This ensures that all tiles contain the same number of iterations, simplifying the process of generating the tiled program.

If H is an optimal tiling, our procedure does not guarantee that H^{-1} is integral. Note that H^{-1} is integral if B^{-1} is integral and $\sqrt[n]{\nu|\det(B)|}$ is an integer. In general, let c be the smallest positive integer such that cB^{-1} is integral. In order for H^{-1} to be integral, we must choose a computation volume from the following set:

$$\{\frac{x^n c^n}{|\det(B)|} \mid x \text{ is an positive integer}\}$$

Consider the optimal tilings in Example 8. Both B_1^{-1} and B_2^{-1} are integral. To make H_1^{-1} and H_2^{-1}

m \ n	3			4			5		
4	0.03:**3**	0.06:**3.9**	0.07:**4**						
5	0.07:**3**	0.10:**3.6**	0.13:**5**	0.17:**4**	0.24:**5.3**	0.33: **6**			
6	0.10:**3**	0.15:**3.7**	0.23:**6**	0.33:**4**	0.60:**6.2**	1.00: **8**	0.33:**5**	1.39: **7.6**	2.50: **9**
7	0.17:**3**	0.21:**3.8**	0.30:**6**	0.50:**4**	1.01:**6.8**	2.50:**10**	1.17:**5**	7.16:**10.0**	28.83:**14**
8	0.23:**3**	0.28:**3.8**	0.33:**5**	0.67:**4**	1.64:**7.3**	3.17:**10**	1.67:**5**	24.72:**12.4**	116.67:**18**
9	0.27:**3**	0.35:**3.9**	0.43:**6**	1.17:**4**	2.42:**7.7**	6.00:**12**	3.33:**5**	55.40:**14.1**	281.17:**21**
10	0.33:**3**	0.45:**3.9**	0.57:**6**	1.83:**4**	3.24:**7.7**	6.83:**12**	4.17:**5**	109.92:**15.8**	590.00:**24**
11	0.43:**3**	0.54:**3.8**	0.67:**6**	2.93:**4**	4.33:**8.1**	11.67:**14**	10.00:**5**	206.47:**17.9**	1610.00:**25**
12	0.50:**3**	0.65:**3.8**	0.83:**7**	3.67:**4**	5.60:**8.3**	12.83:**14**	16.67:**5**	279.26:**18.2**	3273.33:**33**

Figure 11: The execution times of the tiling procedure in milliseconds. $\boxed{t_b{:}\mathbf{r_b}}\boxed{t_a{:}\mathbf{r_a}}\boxed{t_w{:}\mathbf{r_w}}$ for each fixed (m, n) was obtained by running the procedure on 100 randomly generated dependence matrices of size $n \times m$ with matrix elements drawn from [−2,2]; where the three data entries represent the best-case, average-case and worst-case results, respectively. In each case, denoted by $t : \mathbf{r}$, t is the execution time of the procedure and $\mathbf{r}$ is the number of rays of the tiling cone.

integral, ν must take values from the set:

$$\{72x^3 \mid x \text{ is an positive integer}\}$$

Alternatively, given an arbitrary computation volume ν, one may approximate an optimal tiling H with a matrix F such that F^{-1} is integral. However, this topic is beyond the scope of the paper.

7 Related Work

This section reviews some existing results on tiling with particular emphasis on those aiming at finding communication-minimal tilings (in the sense of this paper).

Pioneering studies on tiling are perhaps those of Irigion and Triolet [11] and Wolfe [21, 22]. Irigion and Triolet formally defined tiling as a loop transformation that divides the iteration space using hyperplanes into parallelepiped tiles and traverses the tiles to cover the iteration space. They also introduced the three important constraints on a tiling: H must be nonsingular, H^{-1} must be integral and $HD \geqslant 0$ must be true. Wolfe demonstrated the feasibility of generating blocked algorithms through strip mining and loop interchanging [22]. This consists of tiling the iteration

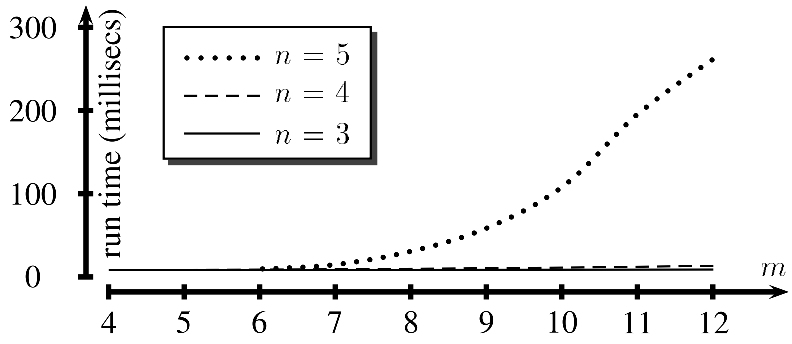


Figure 12: The average run time on finding optimal tilings for dependence matrices $D \in \mathbb{Z}^{n \times m}$.

space with rectangles using a diagonal tiling matrix:

$$H = \text{diag}(h_{1,1}, \cdots, h_{n,n})$$

and is not optimal in general. Since a tiling matrix H must satisfy $HD \geqslant 0$, this simple approach often breaks down when the dependence matrix D contains negative entries. To alleviate this problem, Wolfe [22] proposed to first restructure (e.g. using the wavefront transformation) a loop nest and then tile the restructured program. In the extreme case along this line, Wolf and Lam [19] proceeded to first transform a loop nest into a *fully permutable loop nest* – a loop nest with dependence matrix $D \geqslant 0$, and then settled with a rectangular tiling of the transformed program. A rectangular tiling is always feasible for a set of fully permutable loops. Wolf and Lam's approach applies to loops with iteration vectors but was not developed to find communication-minimal tilings.

Next, we consider three recent papers on finding tilings with a minimal amount of communication. Schreiber and Dongarra were perhaps the first investigating compiler techniques for finding communication-minimal tilings [17]. In their two-step approach, they first determined the shape of a tile by minimising the ratio of the computation volume of a tile to the surface area of a tile and then attempted to adjust the aspect ratios of a tile in order to minimise the amount of local memory and communication induced by a tile. Schreiber and Dongarra formulated the problem of finding the optimal shape of a tile as follows:

Maximise $|\det(H)|$
Subject to The rows of H all have unity Euclidean norm
$HD \geqslant 0$

Essentially, the problem is to find a matrix H that has the largest determinant, subject to $HD \geqslant 0$. Unfortunately, the search space $\mathcal{C}(h)$ contains an infinite number of tiling matrices to be considered. In a heuristics-based procedure, they first generated all tiling matrices whose rows are extremal rays of the tiling cone $\mathcal{C}(h)$ and then applied an orthogonalisation process in an attempt to maximise their determinants. While a tiling matrix may get its determinant increased, some of its rows may no longer be extremal rays. Therefore, this method does not yield communication-minimal tilings. For the dependence matrix in Example 2, $\mathcal{C}(h)$ contains two (normalised) extremal rays $(0,1)$ and $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$. So there is only one tiling matrix:

$$\begin{bmatrix} 0 & 1 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

whose determinant is $-\frac{1}{\sqrt{2}}$. Using the orthogonalisation process in [17, Section 3], the two tiling matrices with orthogonal rows are found:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

both of which have unity determinant. Scaling these two matrices to obtain the tilings with the computation volume ν yields:

$$H_1 = \sqrt{\frac{1}{\nu}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad H_2 = \sqrt{\frac{1}{\nu}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Both tilings are not optimal. It can be checked that $\mathcal{V}_{\text{comm}}(H_1) = 3\sqrt{\nu}$ and $\mathcal{V}_{\text{comm}}(H_2) = 2\sqrt{2}\sqrt{\nu}$, while the optimal tiling given in Example 4 has the communication volume $2\sqrt{\nu}$.

There is a simple reason why Schreiber and Dongarra failed to find optimal tilings. By normalising the rows of every tiling matrix, they explicitly restricted their search for optimal tilings to those each of which induces the same surface area on all all faces of a tile. As shown in Lemma 2, *a tiling that is optimal must induce the same amount of communication not the same surface area on all faces of the tile*. By further using Lemma 5, the problem of finding optimal tilings $H = \sqrt[n]{\frac{1}{\nu|\det(B)|}}\, B$ should be formulated as follows:

Maximise	$\lvert\det(B)\rvert$
Subject to	$B \in G'$ (as in (12))
	$BD \geqslant 0$

which can be solved analytically since G' contains only a finite number of elements.

Ramanujam and Sadayappan [15] required a tiling matrix H to be a lower triangular unimodular matrix. Thus, they solved a simplified version of our optimisation problem (3):

Minimise	$\mathcal{V}_{\text{comm}}(H) = \frac{1}{\lvert\det(H)\rvert} \sum_{i=1}^{n} \sum_{j=1}^{m} (HD)_{i,j}$
Subject to	$HD \geqslant 0$

Since H is unimodular, $\frac{1}{|\det(H)|}$ can be removed from the objective function, rendering the problem a form of integer programming. The optimal solution found is scaled to obtain a tile of an appropriate size. In general, the optimality of this method is not guaranteed. For the dependence matrix in Example 5, the optimal solution to the above problem is the identity matrix. So the optimal tiling with the computation volume ν is:

$$\sqrt{\frac{1}{\nu}}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

which yields the communication volume $8\sqrt{\nu}$, larger than the communication volume $2\sqrt{15}\sqrt{\nu}$ induced by the optimal tiling in Example 5.

The work in this paper drew its inspiration mainly from a recent work by Boulet, et al [3]. They found optimal tilings in two steps. In the first step, the optimal solutions are found to:

$$\begin{array}{rl} \textbf{Minimise} & \frac{1}{|\det(H)|^{\frac{1}{n}}}\sum_{i=1}^{n}\sum_{j=1}^{m}(HD)_{i,j} \\ \textbf{Subject to} & |\det(H)| \neq 0 \\ & HD \geqslant 0 \end{array} \tag{13}$$

In the second step, the solutions are scaled to obtain the tiles with an appropriate size. Our problem formulation (3) is similar but with a linear objective function. This enables us to derive closed-form optimal solutions in several important special cases based only on the inequality of arithmetic and geometric means and to develop the optimal solutions in the general case based primarily on that inequality. In addition, several important issues are addressed in this paper. We proposed to use the echelon reduction to construct the extremal rays of the tiling cone from the dependence cone, which has proved to be very efficient for practical applications. We discussed in detail the dual relationship between the dependence cone and the tiling cone. Finally, we gave an implementation demonstrating the efficiency of our tiling procedure. It is expected that this conceptually simpler framework can provide new insights into tackling other problems in tiling nested loops. The other aspects of this work in relation to that work was already discussed at the beginning of the paper.

Finally, several researchers have studied tiling in the context of compiling programs for distributed memory machines, possibly with user-specified data decomposition directives [12, 14, 16].

8 Conclusion

Inspired by the work [3] and building on the work [17], this paper described a different approach to finding optimal tilings of iteration spaces with a minimal amount of communication through the faces of a tile. The key observation is that a tiling that is optimal must induce the same amount of communication on all faces of a tile, which reduces the search space for optimal tilings to a finite set of matrices whose rows are extremal rays in the tiling cone. For nested loops with several special forms of dependences, closed-form optimal tilings were provided. In the general case, a procedure was given that is guaranteed to always find optimal tilings. An efficient implementation of the procedure was also discussed. The idea of using the echelon reduction to construct the extremal rays from the dependence cone was discussed , and the advantages of this approach were explained and validated by experiments. The experimental results demonstrated that our tiling procedure is very efficient for practical applications. Where appropriate, the geometric insights behind optimal tilings were explained. In particular, the dual relationship between the dependence cone and the tiling cone was exposed. Several existing results were compared and contrasted in detail.

The problem of finding optimal tilings is a difficult non-linear combinatorial problem. But the developments of almost all results in the paper were conducted in a conceptually simple framework, based primarily on the inequality of arithmetic and geometric means and several basic concepts from convex cones. Motivated by the insights provided by this framework, we intend to pursue one important related problem of tiling nested loops to improve cache locality. Some earlier work in this area can be found in [5, 9, 13, 20]

9 Acknowledgements

I would like to thank all referees for their comments and suggestions. I also want to thank Referee B for pointing out a mistake in the formulation of a lemma in the original version of the paper, which has led to a split of that lemma into Lemmas 4 5, and 6 in this paper. This has clarified and refined some results described in the three lemmas.

This work is supported by an Australian Research Council Grant A49600987.

References

[1] U. Banerjee. *Loop Parallelization*. Kluwer Academic Publishers, 1994.

[2] E. F. Beckenbach and R. Bellman. *Inequalities*. Springer-Verlag, 2nd edition, 1965.

[3] P. Boulet, A. Darte, T. Risset, and Y. Robert. (Pen)-ultimate tiling. *Integration, the VLSI Journal*, 17:33–51, 1994.

[4] S. Carr and K. Kennedy. Compiler blockability of numerical algorithms. In *Supercomputing '92*, pages 114–124, Minneapolis, Minn., Nov. 1992.

[5] K. Cooper, K. Kennedy, and N. McIntosh. Cross-loop reuse analysis and its application to cache optimizations. In *Proc. of the 9th Workshop on Languages and Compilers for Parallel Computing*, Aug. 1996.

[6] E. W. Dijkstra. *Predicate Calculus and Programming Semantics*. Series in Automatic Computation. Prentice-Hall, 1990.

[7] J. J. Dongarra, S. J. Hammarline, and D. C. Sorensen. Block reduction of matrices to condensed forms for eigenvalue computations. *J. of Computer Application and Mathematics*, 27:216–227, 1989.

[8] K. Gallivan, W. Jalby, U. Meier, and A. H. Sameh. Impact of hierarchical memory systems on linear algebra algorithm design. *Int. J. of Supercomputer Applications*, 2:12–48, 1988.

[9] G. R. Gao, V. Sarkar, and S. Han. Locality analysis for distributed shared-memory multiprocessors. In *Proc. of the 9th Workshop on Languages and Compilers for Parallel Computing*, Aug. 1996.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins, 2nd edition, 1989.

[11] F. Irigoin and R. Triolet. Supernode partitioning. In *Proc. of the 15th Annual ACM Symposium on Principles of Programming Languages*, pages 319–329, San Diego, California., Jan. 1988.

[12] C. King and L. Ni. Grouping in nested loops for parallel execution on multicomputers. In *Proc. of Int. Conf. on Parallel Processing*, volume 2, pages II–31—II–38, Aug. 1989.

[13] M. S. Lam, E. E. Rothberg, and M. E. Wolf. The cache performance and optimizations of blocked algorithms. In *Proc. of the 2nd International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 63–74, Santa Clara, California, Apr. 1991.

[14] H. Ohta, Y .Saito, M. Kainaga, and H. Ono. Optimal tile size adjustment in compiling for general DOACROSS loop nests. In *Supercomputing '95*, pages 270–279. ACM Press, 1995.

[15] J. Ramanujam and P. Sadayappan. Tiling multidimensional iteration spaces for multicomputers. *J. of Parallel and Distributed Computing*, 16(2):108–230, Oct. 1992.

[16] A. Rogers and K. Pingali. Compiling for distributed memory architectures. *IEEE Transactions on Parallel and Distributed Systems*, 5(3):281–298, Mar. 1994.

[17] R. Schreiber and J. J. Dongarra. Automatic blocking of nested loops. Technical Report 90.38, RIACS, May 1990.

[18] A. Schrijver. *Theory of Linear and Integer Programming*. Series in Discrete Mathematics. John Wiley & Sons, 1986.

[19] M. Wolf and M. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Trans. on Parallel and Distributed Systems*, 2(4):452–471, Oct. 1991.

[20] Michael E. Wolf, , and Monica S. Lam. A data locality optimizing algorithm. In *Proc. of the ACM SIGPLAN'91 Conf. on Programming Language Design and Implementation*, Jun. 1991.

[21] M. J. Wolfe. Iteration space tiling for memory hierarchies. In G. Rodrigue, editor, *Parallel Processing for Scientific Computing*, pages 357–361, Philadelphia PA, 1987.

[22] M. J. Wolfe. More iteration space tiling. In *Supercomputing '88*, pages 655–664, Nov. 1989.

[23] M. J. Wolfe. *Optimizing Supercompilers for Supercomputers*. Research Monographs in Parallel and Distributed Computing. MIT Press, 1989.

[24] M. J. Wolfe. *High Performance Compilers for Parallel Computing*. Addision-Wesley, 1996.

[25] J. Xue. On tiling as a loop transformation. In *Proc. of the SPDP Workshop on Challenges in Compiling for Scalable Parallel Systems*, New Orleans, 1996. IEEE Computer Society Press.

[26] Y.Q. Yang, C. Ancourt, and F. Irigoin. Minimal data dependence abstractions for loop transformations. In *Proc. of the 7th Workshop on Languages and Compilers for Parallel Computing*, Ithaca, Aug 1994.