Journal of VLSI Signal Processing, , 1–16 (1998)
© 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Parametric Analysis of Polyhedral Iteration Spaces

PHILIPPE CLAUSS AND VINCENT LOECHNER

{*clauss,loechner*}*@icps.u-strasbg.fr*

ICPS, Université Louis Pasteur, Strasbourg, Pôle API, Bd. Sébastien Brant, 67400 Illkirch, France

Received March 30, 1997. Revised January 10, 1998.

Editors: V. Taylor and K. Parhi

Abstract. In the area of automatic parallelization of programs, analyzing and transforming loop nests with parametric affine loop bounds requires fundamental mathematical results. The most common geometrical model of iteration spaces, called the *polytope model*, is based on mathematics dealing with convex and discrete geometry, linear programming, combinatorics and geometry of numbers.

In this paper, we present automatic methods for computing the parametric vertices and the *Ehrhart polynomial*, i.e. a parametric expression of the number of integer points, of a polytope defined by a set of parametric linear constraints.

These methods have many applications in analysis and transformations of nested loop programs. The paper is illustrated with exact symbolic array dataflow analysis, estimation of execution time, and with the computation of the maximum available parallelism of given loop nests.

Keywords: Integer Points, Parametric Polytopes, Ehrhart Polynomials, Parametric Vertices, Automatic Parallelization, Nested loops, Optimizing Compilers.

1. Introduction

From systolic algorithm synthesis to nested loops parallelization, a common geometric model is used, called the *polytope model*, leading to common fundamental mathematical problems, dealing with discrete and convex geometry, combinatorics, geometry of numbers and linear programming.

One of these fundamental problems is the counting of integer points, i.e. points having integer coordinates, contained in a polytope. Moreover, this counting has to be parameterized, since scientific algorithms often depend on size parameters. Such a counting is necessary when analyzing a parallel algorithm modeled by a polyhedral iteration space. Many kinds of analysis reduce to this problem: finding the number of flops executed by a loop, the number of memory locations touched by a loop, the communication volume, the maximum parallelism, and so forth. The resulting information can be used to estimate the execution time of a code segment, compare the memory bandwidth requirements vs. the flops count of a code segment [1], determine which loops will flush the cache, and then calculate the cache miss rate [2], [3], evaluate message traffic and allocate message buffers, determine whether a parallel loop is load balanced (i.e., performs the same amount of computation in each iteration [4]), or if not balanced, determine number of iterations to assign to each processor in order to balance the work load (balanced chunk-scheduling) [5], exploit maximum parallelism, optimize the number of needed processors, and optimize loop and data partitionning.

Another fundamental problem is to determine the parametric bounds of a given index, or of a given affine combination of indices, induced by a nested loop program. Finding the parametric latency of a parallel loop, the parametric bounds of a loop scanning a polyhedral space [6], or the parametric indices of the last iteration updating an array element used during a given one (*exact symbolic array dataflow analysis*) [7], are all prob-

lems that can reduce to find the domains associated to stable sets of parametric vertices.

Generally, parametric analysis can be opportune even for non-parameterized problems: for example, analysis of a perfect loop nest parallelization can be achieved by considering execution steps as parameters.

Such informations are essential to optimizing compilers when transforming a parallel program.

In this paper, we present an algorithm for computing the parametric vertices and their associated domains of a convex polytope defined by a set of parametric linear constraints, and an automatic method for computing an exact and closed expression of the number of integer points contained in such a convex polytope or in a union of such convex polytopes. This counting method consists of first computing the parametric vertices, and then computing the *Ehrhart polynomial*, i.e. a parametric expression of the number of integer points.

This second part is based on results due to the mathematician E. Ehrhart, who solved this problem for parametric polytopes that depend on one positive integral parameter. We propose an extension of his results to any number of parameters, showing that this extension is made possible by our new parametric vertices finding algorithm.

This algorithm is based on the classical representation of a parametric polytope as a non-parametric polyhedron in the combined parameter and data space. The computation of the faces whose dimension equals the number of parameters, allows the expression of the parametric vertices as affine functions of the parameters. A validity domain partition for the vertices is also computed: each subdomain corresponds to a certain set of vertices of the parametric polytope.

The paper is organized as follows. Section 2 is dedicated to the mathematical background. In section 3, the algorithm to find the parametric vertices is described. It makes possible, in section 4, an extension of Ehrhart's results on the parametric counting of integer points. Finally, we illustrate the methods in section 5 by an example of exact symbolic array dataflow analysis, in section 6 by counting the number of flops executed for a given loop nest, and in section 7 by computing the parallelism rate for a given iteration space and a given linear parallelization. This rate is parameterized by the size parameter and the execution step. In section 8, conclusions and comparisons with related works are given.

2. Definitions and assumptions

We first recall some basic notions dealing with geometry of numbers [8], [9], [10] and enumerative combinatorics [11], [12]. Then some more specific concepts, closely related to the content of the paper, are introduced.

2.1. Polyhedra and polytopes

Let $\mathbb{Q}$ denote the set of rational numbers and $\mathbb{Z}$ the set of integers. A convex polyhedron is defined by a finite set of linear inequalities:

$$P = \{x \in \mathbb{Q}^d \mid A \cdot x \leq b\}$$

where A is a rational matrix and b a rational vector. This is called the *implicit representation* of the polyhedron. A bounded convex polyhedron is called a convex polytope, and represents the convex hull of a finite number of points $x_1, \ldots, x_{m+1}$ in $\mathbb{Q}^d$. If $x_1, \ldots, x_{m+1}$ are affinely independent points of $\mathbb{Q}^d$ (which also means that $x_2 - x_1$, $x_3 - x_1, \ldots, x_{m+1} - x_1$ are linearly independent) then the convex hull of $x_1, \ldots, x_{m+1}$ is called an m-dimensional *simplex* of $\mathbb{Q}^d$ with vertices $x_1, \ldots, x_{m+1}$.

Since our goal is to count the number of integer points contained in a polyhedron, only polytopes are considered.

The *affine span* of P is the smallest affine subspace of $\mathbb{Q}^d$ entirely containing P. The *dimension* of P is the dimension of its affine span, and a k-dimensional polytope is called a k-polytope for brevity.

DEFINITION 1 (geometric representation) *A convex polytope P has a dual geometric representation as a combination of vertices:*

$$P = \{x \in \mathbb{Q}^d \mid x = V\nu, \quad \nu_i \geq 0, \quad \sum \nu_i = 1\}$$

where the columns of matrix V are the coordinates of its vertices and ν is a free vector with elements ν_i.

Any polytope P has a unique minimal geometric representation. Procedures exist to compute the dual representations of P, that is, given A and b in the implicit form, compute V and vice versa (see [13] for references).

If $\vec{c}$ is a nonzero vector and γ is defined by $\gamma = MAX\{\vec{c} \cdot x \mid x \in P\}$, the affine hyperplane $\{x \mid \vec{c} \cdot x = \gamma\}$ is called a supporting hyperplane of P. The *faces* of P are $\emptyset$, P, and the intersections of P with its supporting hyperplanes. Each face of P is itself a polyhedron and is called a k-face if it is k-dimensional. If P is a k-polyhedron, faces of P of dimension 0, 1 and $k-1$ are called *vertices, edges* and *facets*, respectively.

In this work, we consider polytopes which may be non-convex, called *normal polyhedra*, according to the following definition. This definition corresponds to the one given by Ehrhart in [14] and [15], p. 47 and includes unions of convex polytopes.

DEFINITION 2 (normal polyhedron) *A normal polyhedron P is a closed polyhedron which admits a finite rectilinear simplicial subdivision. It is then the union of a finite number of simplices $S_1, S_2, ..., S_n$ of same dimension such that,*

- *The intersection between two simplices is either empty or is a common face,*
- *Each facet of one of these simplices belongs to at most one other simplex, faces belonging to only one simplex being the border of the polyhedron,*
- *These simplices can be ordered such that for any $h < n$, if $P_h = S_1 \cup S_2 \cup ... \cup S_h$, the intersection $P_h \cap S_{h+1}$ is either empty, or is the union of common facets of S_{h+1} and the border of P_h.*

2.2. Parameterized lattice polytopes

A parameterized convex polytope is defined by:

$$P_N = \{x \in \mathcal{Q}^d \mid A \cdot x \leq B \cdot N + c\}$$

where N is the symbolic p-vector of the parameters, A and B are constant rational matrices of size $(m \times d)$ and $(m \times p)$ respectively, and c is a constant rational m-vector.

A lattice is the set of points $\mathcal{L}^d = \{\alpha_1\vec{g}_1 + ... + \alpha_d\vec{g}_d \mid \alpha_1, ..., \alpha_d \in \mathcal{Z}\}$ where $\vec{g}_1, ..., \vec{g}_d$ are linearly independent vectors of $\mathcal{Q}^d$, called the basis of the lattice. The set of all the integral points is called *standard lattice* and denoted by $\mathcal{Z}^d$. Let G be the matrix whose columns are the vectors $\vec{g}_1, ..., \vec{g}_d$, called the *dilatation matrix*. Any lattice $\mathcal{L}^d$ consists of the points $G \cdot z$, $z \in \mathcal{Z}^d$, i.e. $\mathcal{L}^d = G \cdot \mathcal{Z}^d$. So we get the most general lattice $\mathcal{L}^d$ by subjecting the standard lattice $\mathcal{Z}^d$ to an arbitrary non-singular transformation G [8]. Hence, without loss of generality, only the standard lattice $\mathcal{Z}^d$ is considered in the rest of the paper. The *enumerator* of any set E is the number of points in $E \cap \mathcal{L}^d$.

The intersection between a lattice $\mathcal{L}^d$ and a polytope is called a lattice polytope P of $\mathcal{L}^d$. A lattice polytope P of $\mathcal{L}^d$ is a $\mathcal{L}^d$-polytope if all of its vertices are in $\mathcal{L}^d$. If $\mathcal{L}^d = \mathcal{Z}^d$, P is an *integral polytope*. If some vertices of P do not belong to $\mathcal{L}^d$ and have rational coordinates, P is a *rational* polytope.

Since by definition, any lattice $\mathcal{L}^d$ can be deduced from $\mathcal{Z}^d$ by a non-singular transformation G, any $\mathcal{L}^d$-polytope is affinely equivalent to an integral polytope. Hence, results given in the rest of the paper and concerning integral polytopes can be directly extended to $\mathcal{L}^d$-polytopes. From now on and without loss of generality, the generic term of integral polytopes will be used.

Our method of counting the number of integer points is based on the decomposition of a parametric polytope into several *homothetic-bordered* systems, associated with *validity domains*, according to the following definition which is an extension of Ehrhart's definition [14] to any number of parameters.

DEFINITION 3 (homothetic-bordered system) *Let H_N, $N = (n_1, n_2, \ldots, n_q)$, be a system defined by constraints of the form $\sum a_i x_i = \sum b_j n_j + c$, $\sum a_i x_i < \sum b_j n_j + c$, $\sum a_i x_i \leq \sum b_j n_j + c$, where the a_i's, the b_j's and the c's are given integers, the x_i's are free variables and the n_j's are positive integral parameters.*

Such a system is homothetic-bordered if and only if the polytope it defines has vertices whose coordinates are affine combinations of the parameters.

In general, multi-parametric polytopes (with several parameters) are described by a collection of homothetic-bordered polytopes, each defined on an adjacent subdomain of parameter values called

the *validity domain*, as it is illustrated by the following example. The determination of these subdomains is done by the parametric vertices finding algorithm described in section 3.

Example: In figure 1, we have three parameterized, one dimensional polytopes,

$$\begin{aligned} P_{n_1} &= \{x \mid x \geq 0, x \leq n, 2x \leq n+6\} \\ P_{n_2} &= \{x \mid x \geq 0, n \geq 0, 2x \leq n+6\} \\ P_{n_3} &= \{x \mid 0 \leq x \leq n\} \end{aligned}$$

P_{n_3} is a homothetic-bordered system, and P_{n_2} is a homothetic-bordered system. P_{n_1} is not homothetic-bordered, but can be made as a union of homothetic-bordered systems qualified by certain subdomains of the parameter (n) space,

$$P_{n_1} = \cup \begin{cases} P_{n_3} & when \;\; 0 \leq n \leq 6 \\ P_{n_2} & when \;\; 6 \leq n \end{cases}$$

[Example due to Doran K. Wilde.] □

2.3. Pseudo-polynomials

In the following, we denote the enumerator of P_N as j_N, i.e. the number of integer points of P_N. It will be formulated using *polynomials* and *pseudo-polynomials* according to the following definitions. We extend Ehrhart's definition of periodic numbers [15] to any dimension.

DEFINITION 5 (periodic number) *A one-dimensional periodic number* $u(n) = [u_1, u_2, \ldots, u_p]_n$ *is equal to the item whose rank is equal to* $n \mod p$, p *is called the period of* $u(n)$.

$$u(n) = \begin{cases} u_1 & if \;\; n = 1 \;(mod\; p) \\ u_2 & if \;\; n = 2 \;(mod\; p) \\ \ldots & \\ u_p & if \;\; n = 0 \;(mod\; p) \end{cases}$$

Let $N = (n_j)$ *be a* q*-dimensional vector,* $j \in [1 \ldots q]$. *A* q*-periodic number* $u(N)$ *is defined by a* q*-dimensional array* $U_{(i_1,i_2,\ldots,i_q)}$ *of size* $p_1 \times p_2 \times \ldots \times p_q$ *such that*

$$u(N) = U_{(i_1,i_2,\ldots,i_q)}$$
$$if \;\; \forall j \in [1 \ldots q], \; n_j = i_j \mod p_j$$

The vector $p = (p_j)$ *is called the multi-period of* $u(N)$ *and the least common multiple of the* p_i*'s is called the period of* $u(N)$.

For example,

- $(-1)^n = [-1, 1]_n$
- $(-1)^{n-m} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}_{(n,m)}$

Ehrhart in [15] gives the basic properties of one-dimensional periodic numbers which can be easily extended to the multi-dimensional case: Let $u(N)$ and $u'(N)$ be q-periodic numbers of multi-period p and p' respectively. They can be reduced to the same multi-period $m = (m_i)$, where $m_i = p_i\alpha = p'_i\beta$ is the least common multiple of p_i and p'_i, by repeating α times (respectively β times) the sequence of items in the i^{th} dimension of $u(N)$ (respectively $u'(N)$). Since $u(N)$ and $u'(N)$ are reduced to the same period, $u(N) + u'(N) = (u_I + u'_I)_N$ and $u(N)u'(N) = (u_I u'_I)_N$.

Likewise, we also extend Ehrhart's definition on pseudo-polynomials.

DEFINITION 6 (pseudo-polynomial) *A polynomial* $f(N)$ *is a pseudo-polynomial if some of its coefficients are periodic numbers instead of constants. The least common multiple of the periods of its periodic coefficients is the pseudo-period of* $f(N)$. *Thus, a pseudo-polynomial has two characteristics, its degree and its pseudo-period.*

3. A parametric vertices finding algorithm

3.1. Parametric vertices

The algorithm presented below is fully explained in [16]. It computes the vertices $v_i(N)$ of a parametric polytope defined by:

$$P_N = \{x \in \mathcal{Q}^d \mid Ax \leq BN + c\}$$

where N is a vector of the form $\begin{pmatrix} n_1 \\ \vdots \\ n_p \end{pmatrix}$, A and B are rational matrices and c is a rational vector.

Parametric polytopes can be represented as non-parametric polyhedra in the *combined* data and parameter space as follows:

$$P' = \left\{ \begin{pmatrix} x \\ N \end{pmatrix} \in \mathcal{Q}^{d+p} \;\middle|\; [A \mid -B] \begin{pmatrix} x \\ N \end{pmatrix} \leq c \right\}$$

The following property relates a combined polyhedron P' to its equivalent parametric polyhedron P_N.

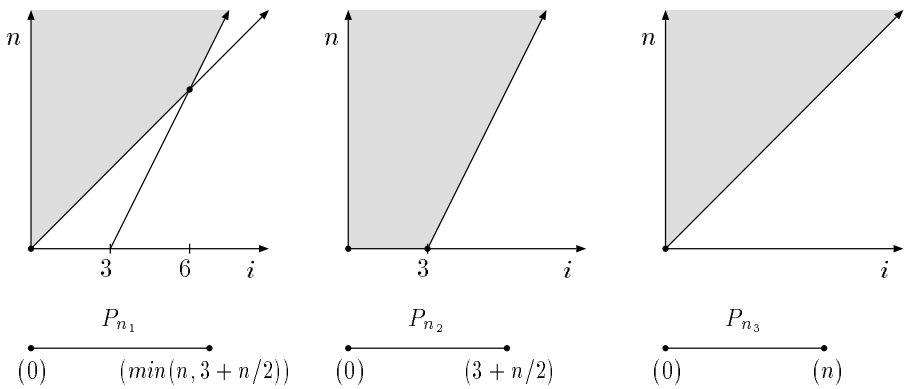

Fig. 1. The parameterized polytopes P_{n_1}, P_{n_2} and P_{n_3}

PROPERTY 1 *Let S_{N_0} be the subspace of $\mathbb{Q}^{d+p}$ associated with a given value of $N = N_0$:*

$$S_{N_0} = \left\{ \begin{pmatrix} x \\ N_0 \end{pmatrix} \in \mathbb{Q}^{d+p} \right\}$$

The intersection of P' with S_{N_0} fixes the parameter to a constant vector N_0. The result is the polyhedron P_{N_0}, extended in the combined space $\mathbb{Q}^{d+p}$:

$$P' \cap S_{N_0} = \{ \begin{pmatrix} x \\ N \end{pmatrix} \in \mathbb{Q}^{d+p} \mid x \in P_{N_0}, N = N_0 \}.$$

P_{N_0} can be obtained by projecting this intersection onto the data space $\mathbb{Q}^d$.

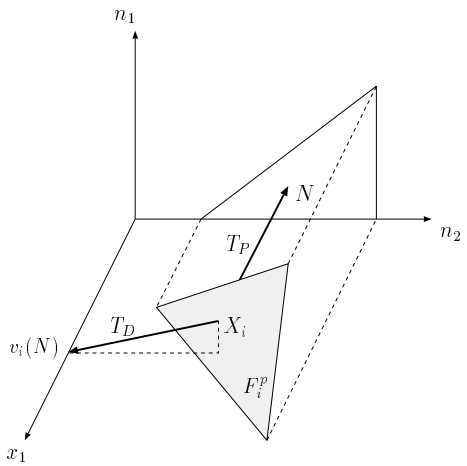

Fig. 2. A p-face F_i^p (highlighted) and the computation of $v_i(N)$ as a function of N.

We assume in the following that there is no equality of the form: $DN + d = 0$ where D is a non-null matrix and d a vector included in the set of constraints defining the parametric polyhedron. If such an equality exists, it can be easily detected and removed. This has to be verified for the following property to be true:

PROPERTY 2 *Let $v_i(N)$ be a vertex of the polyhedron P_N. There exists a p-face of P' denoted F_i^p such that $v_i(N)$ corresponds to $F_i^p \cap S_N$.*

Proof: The detailed proof can be found in [16]. It is based on the definition of $v_i(N)$ which is a 0-face of P_N: $v_i(N)$ is the intersection of a hyperplane $\mathcal{H}_i$ with P_N. In the combined space, $\mathcal{H}_i$ is the intersection of S_N with a tangent hyperplane of P' denoted $\mathcal{H}'_i$. $\mathcal{H}'_i$ can be chosen such that $\mathcal{H}'_i \cap P'$ is a p-face of P' and the result follows. ■

Let Π be the function projecting the points in $\mathbb{Q}^{d+p}$ onto the parameter space $\mathbb{Q}^p$: $\Pi \begin{pmatrix} x \\ N \end{pmatrix} = N$.

PROPERTY 3 *The vertex $v_i(N)$ corresponding to face F_i^p is defined only for the values of N contained in the polyhedron $\Pi(F_i^p)$.*

Proof: The range of N over which $F_i^p \cap S_N \neq \emptyset$ is the projection of F_i^p on the parameter space $\mathbb{Q}^p$. Since $\begin{pmatrix} v_i(N) \\ N \end{pmatrix} = F_i^p \cap S_N$, the validity domain for $v_i(N)$ is $N \in \Pi(F_i^p)$. $\Pi(F_i^p)$ is a convex polyhedron since F_i^p is a convex polyhedron. ■

Let F_i^p be a p-face of P'. If F_i^p corresponds to a vertex $v_i(N)$ then the intersection of F_i^p with S_N is the single point $\begin{pmatrix} v_i(N) \\ N \end{pmatrix}$. Let $\mathcal{B}$ be a basis of the supporting affine space of F_i^p and X_i the coordinates in basis $\mathcal{B}$ of $\begin{pmatrix} v_i(N) \\ N \end{pmatrix}$. We can compute the affine transformations T_D and T_P such that:

$$N = T_P(X_i) \quad (1)$$
$$v_i(N) = T_D(X_i) \quad (2)$$

Transformation T_P is bijective if and only if no equality of the form $\mathcal{L}(p) = K$ is valid on F_i^p where $\mathcal{L}$ is a linear and non-null function.

If T_P is bijective, the intersection of F_i^p with S_N is a single point (cf. figure 2), and F_i^p corresponds to a vertex of P_N. Equation (1) can be rewritten as: $T_P^{-1}(N) = X_i$. And by replacing X_i by this value in equation (2) we find:

$$v_i(N) = T_D(T_P^{-1}(N)).$$

Conversely, if T_P is not bijective, then there does not exist one single point in F_i^p corresponding to a given value of N, but a whole k-polyhedron, with $k > 0$. The intersection of this polyhedron with S_N does not correspond to a vertex of P_N, but to a k-face of P_N.

Finally, the algorithm consists of:

1. compute all faces F_i^p,
2. for each face F_i^p:
   - compute T_D and T_P,
   - if T_P is non-singular, compute $T_D \circ T_P^{-1}$, report this vertex, and compute $\Pi(F_i^p)$ defining the corresponding set of constraints on N.

3.2. Partitioning the parameter space

The output of the above algorithm is a set of vertices and for each vertex, a set of constraints that must be verified by the parameters N for this vertex to be valid.

The algorithm presented below computes the set of distinct parameter subdomains called *validity domains*. The validity domains separate complete sets of parametric vertices: there will be one single homothetic or bordered polytope defined on each of these validity domains, as proved in the next section.

The validity domains are obtained by making a partition of the parameter space from the projection of the p-faces of P'. Hence, for each N inside a given validity domain, the set of p-faces intersected by S_N is the same. Therefore the set of parametric vertices of P_N is also the same.

The whole parameter validity domain of P_N is $\Pi(P')$, the projection of P' onto the parameter space $\mathcal{Q}^p$. The projection of the convex p-faces of P' forms a set of polyhedra in $\mathcal{Q}^p$ that covers $\Pi(P')$, since P' is entirely defined by its set of p-faces. The algorithm described below computes a set of polyhedra $D = \{d_j \mid 1 \leq j \leq v\}$ where v is the number of validity domains, such that:

- d_j is a polyhedron of dimension p.
- $d_j \cap d_k$ is a polyhedron of dimension lower than p, $\forall j \neq k$.
- $d_1 \cup d_2 \cup \ldots \cup d_f = \Pi(P')$
- $\forall N \in d_j, S_N \cap D'$ yields a unique set of parametric vertices.

The algorithm constructs the partition D by adding successively each projected face $f = \Pi(F_i^p)$ to the set of validity domains. At each step, a new partition $NewD$ is defined as:

1. all the intersections $d_j \cap f$ with each of the domains $d_j \in D$,
2. all the differences $d_j - f$, with $d_j \in D$,
3. the difference between f and the union of all domains $d_j \in D$:

$$f - \left(\bigcup_{d_j \in D} d_j \right).$$

PROPERTY 4 *The domains constructed during the intermediate computations by this algorithm may be non convex, however, when the algorithm ends, each resulting validity domain is convex.*

Proof: When the algorithm ends, each resulting validity domain can be described as a finite intersection of the projections of a set of p-faces, corresponding to a finite set of parameterized vertices: the domain corresponding to this set of parameterized vertices can not be cut by the algorithm, since this would imply that there exists a p-face cutting this domain, and the set of parameterized

vertices would not be unique on this validity domain.

The projections of the p-faces are convex, since the projection of a rational convex polyhedron is a rational convex polyhedron, and the p-faces are convex. It follows that the final validity domains are finite intersections of convex sets, which are convex. ■

Construction of the validity domains

Input: A set of faces F^p.
Output: A set of validity domains D.

begin
  $D = \{\}$
  for all $F_i^p \in F^p$ **do**
    $f = \Pi(F_i^p)$
    $NewD = \{\}$
    for $j = 1$ **to** $Card(D)$ **do**
      if $Dimension(d_j \cap f) = p$ **then**
        $NewD = NewD \cup \{d_j \cap f\} \cup \{d_j - f\}$
        $f = f - d_j$
      else
        $NewD = NewD \cup \{d_j\}$
      endif
    done
    if $f \neq \emptyset$ **then**
      $NewD = NewD \cup \{f\}$
    endif
    $D = NewD$
  done
end

These validity domains are essential for the extension of Ehrhart's results presented in the following section.

4. Parametric counting through Ehrhart polynomials extension

In this section, we recall and extend some results due to E. Ehrhart who made a study of the enumeration of lattice points in polyhedra, by means of polynomials along the same lines as Euler. All the initial proofs can be found in [17], [14], [18], [15]. Some aspects of Ehrhart's work were corrected, streamlined, and expanded by I. G. Macdonald [19], [20] and R. P. Stanley [11], [12].

DEFINITION 7 (denominator) *The denominator of a rational point is the lowest common multiple of the denominators of its coordinates. The denominator of a rational polyhedron is the least common multiple of the denominators of its vertices.*

Ehrhart's main result consists in the following theorem dealing with parametric polytopes which depend on only one parameter n.

THEOREM 1 (Ehrhart's fundamental theorem) *The enumerator j_n of any homothetic-bordered k-polyhedron P_n is a polynomial in n of degree k if P_n is integral, and it is a pseudo-polynomial in n of degree k whose pseudo-period is the denominator of P_n if P_n is rational.*

We extend this theorem to any number of parameters $N = (n_1, \ldots, n_p)$.

THEOREM 2 (fundamental theorem extended) *The enumerator j_N of any k-polytope P_N is expressed at different domains of the parameters values by different multi-variable polynomials in N of degree k^p if P_N is integral, and by different multi-variable pseudo-polynomials in N of degree k^p whose pseudo-period is the denominator of P_N if P_N is rational. The parameter validity domains give the parameter ranges where the parametric vertices of P_N are defined. These polynomials or pseudo-polynomials are of the form:*

$$j_N = \sum_{i_1=0}^{k} \sum_{i_2=0}^{k} \cdots \sum_{i_p=0}^{k} c_{i_1,i_2,\ldots,i_p} n_1^{i_1} n_2^{i_2} \ldots n_p^{i_p}$$

where the $c_{i_1,i_2,\ldots,i_p}$'s are periodic numbers whose dimensions are at most p.

Proof of Theorem 2: Any parametric polytope is first decomposed into a collection of homothetic-bordered polytopes defined on adjacent domains of the parameter values. Given such a polytope $P_N = P_{n_1,n_2,\ldots,n_p}$ being part of this decomposition, we have to prove that its number of integer points is a multi-variable pseudo-polynomial in N of degree k^p.

By direct application of Ehrhart's results, $P_{n_1,0,0,\ldots,0}$ is a homothetic-bordered polytope depending on one parameter n_1, whose number of integer points is a pseudo-polynomial in n_1 of degree k. The same reasonning can be applied to

$P_{n_1,1,0,\ldots,0}$, $P_{n_1,2,0,\ldots,0}$, ...which can be summarized in $\forall\, n_2 \in N, j_{n_1,n_2,0,\ldots,0} = c_k n_1^k + c_{k-1} n_1^{k-1} + \ldots + c_0$, where any value c_i is depending on n_2.

Application of the same reasonning to $P_{0,n_2,0,\ldots,0}$, $P_{1,n_2,0,\ldots,0}$, $P_{2,n_2,0,\ldots,0}$, ...yields $\forall\, n_1 \in N, j_{n_1,n_2,0,\ldots,0} = c'_k n_2^k + c'_{k-1} n_2^{k-1} + \ldots + c'_0$, where any value c'_i is depending on n_1.

Hence, we have $c_k n_1^k + c_{k-1} n_1^{k-1} + \ldots + c_0 = c'_k n_2^k + c'_{k-1} n_2^{k-1} + \ldots + c'_0$, showing that any c_i is a pseudo-polynomial in n_2 of degree k and that any c'_i is a pseudo-polynomial in n_1 of degree k. Thus the theorem is proven for any homothetic-bordered polytope depending on two parameters.

Consider now $P_{n_1,n_2,0,0,\ldots,0}$, $P_{n_1,n_2,1,0,\ldots,0}$, $P_{n_1,n_2,2,0,\ldots,0}$, ..., such polytopes are homothetic-bordered polytopes depending on two parameters. Thus, as proven above, their number of integer points is a pseudo-polynomial in n_1 and n_2 of degree k^p. This can be summarized in $\forall\, n_3 \in N, j_{n_1,n_2,n_3,0,\ldots,0} = c_{k,k} n_1^k n_2^k + c_{k,k-1} n_1^k n_2^{k-1} + \ldots + c_0$, where any value $c_{i,j}$ is depending on n_3.

Application of the same reasonning to $P_{0,0,n_3,0,\ldots,0}$, $P_{1,0,n_3,0,\ldots,0}$, $P_{0,1,n_3,0,\ldots,0}$, $P_{1,1,n_3,0,\ldots,0}$, $P_{2,1,n_3,0,\ldots,0}$, ...yields $\forall\, n_1, n_2 \in N, j_{n_1,n_2,n_3,0,\ldots,0} = c'_k n_3^k + c'_{k-1} n_3^{k-1} + \ldots + c'_0$, where any value c'_i depends on both n_1 and n_2.

Hence, we have $c_{k,k} n_1^k n_2^k + c_{k,k-1} n_1^k n_2^{k-1} + \ldots + c_0 = c'_k n_3^k + c'_{k-1} n_3^{k-1} + \ldots + c'_0$, showing that any $c_{i,j}$ is a pseudo-polynomial in n_3 of degree k and that any c'_i is a pseudo-polynomial in n_1 and n_2 of degree k^2. Thus the theorem is proven for any homothetic-bordered polytope depending on three parameters.

This reasonning can be applied in the same way and indefinitely to any number of parameters. Hence, the theorem is proven: the number of integer points of any homothetic-bordered polytope P_N is a multi-variate pseudo-polynomial in N of degree k^p. ■

Due to these theorems, the form of the parametric counting formulae is known: the period is equal to the denominator of the k-polyhedron P_N and the degree is equal to k. Hence, the determination of the Ehrhart pseudo-polynomial can be done by counting some initial values of the enumerator and solving systems of symbolic rational linear equations. Since the number of equations of the system must be equal to the number of unknown values, the complexity depends on the number of parameters, the number of validity domains, the dimension, and the denominator.

The number of unknowns can be reduced using the following theorem, conjectured by Ehrhart and proved by Stanley.

THEOREM 3 *Let P_n be a rational k-polytope and let the Ehrhart pseudo-polynomial be*

$$j_n = c_k(n) n^k + c_{k-1}(n) n^{k-1} + \ldots + c_0(n)$$

where $c_0, \ldots, c_k$ are periodic numbers of n. If for some $q \in [0,k]$, the affine span of every q-dimensional face of P_1 contains a lattice point, then for any $p \geq q$, $c_p(n)$ is constant, i.e. of period one.

Example: Consider the set of linear constraints $\{0 \leq i \leq \frac{1}{2}n, 0 \leq j \leq \frac{1}{2}m\}$ defining the rational 2-polytope $P_{n,m}$,

$$P_{n,m} = \{\binom{i}{j} \in \mathcal{Z}^2 \mid \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \binom{i}{j} \leq \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \binom{n}{m}\}$$

$P_{n,m}$ is the closed rectangle whose parametric vertices are $(0,0)$, $(\frac{n}{2},0)$, $(0,\frac{m}{2})$, $(\frac{n}{2},\frac{m}{2})$.

By looking at $P_{1,0}$, we observe that one vertex, $(\frac{1}{2},0)$, is not an integer point. Hence, the coefficient of n^0 is periodic. The other coefficients are constant, since all the edges, and obviously the polytope itself, contain at least one integer point.

By looking at $P_{0,1}$, we observe that one vertex, $(0,\frac{1}{2})$, is not an integer point. Hence, the coefficient of m^0 is periodic. The other coefficients are constant, since all the edges, and obviously the polytope itself, contain at least one integer point.

Moreover, the period of the periodic coefficients is equal to the denominator of $P_{n,m}$, i.e. 2. The pseudo-polynomial $j_{n,m} = c_1 n^2 + c_2 n + [c_3, c_4]$ can thus be determined by solving the following system of rational linear equations:

$$\begin{cases} c_4 & = j_{0,m} \\ c_1 + c_2 + c_3 & = j_{1,m} \\ 4c_1 + 2c_3 + c_4 & = j_{2,m} \\ 9c_1 + 3c_2 + c_3 & = j_{3,m} \end{cases}$$

$j_{0,m}, j_{1,m}, j_{2,m}$ and $j_{3,m}$ have to be computed. For example, we compute $j_{0,m} = c_5 m^2 + c_6 m + [c_7, c_8]$:

$$\begin{cases} c_8 & = j_{0,0} \\ c_5 + c_6 + c_7 & = j_{0,1} \\ 4c_5 + 2c_6 + c_8 & = j_{0,2} \\ 9c_5 + 3c_6 + c_7 & = j_{0,3} \end{cases}$$

From $P_{0,0}$ to $P_{0,3}$, we count $j_{0,m} = 1, 1, 2, 2$. It follows that $j_{0,m} = \frac{1}{2}m + [\frac{1}{2}, 1]$

Finally, $j_{n,m}$ is determined:

$$\begin{aligned} j_{n,m} &= \left(\tfrac{1}{4}m + \left[\tfrac{1}{4}, \tfrac{1}{2}\right]_m\right) n \\ &\quad + \left[\tfrac{1}{4}m + \left[\tfrac{1}{4}, \tfrac{1}{2}\right]_m, \tfrac{1}{2}m + \left[\tfrac{1}{2}, 1\right]_m\right]_n \\ &= \tfrac{1}{4}nm + \left[\tfrac{1}{4}, \tfrac{1}{2}\right]_m n + \left[\tfrac{1}{4}, \tfrac{1}{2}\right]_n m \\ &\quad + \begin{bmatrix} 1/4 & 1/2 \\ 1/2 & 1 \end{bmatrix}_{n,m} \end{aligned}$$

This parametric formula can be evaluated for any value of n and m. For example, the number of integer points in $P_{4500,2001}$ is determined by $j_{4500,2001} = \frac{1}{4} \times 4500 \times 2000 + \frac{1}{4} \times 4500 + \frac{1}{2} \times 2001 + \frac{1}{2} = 2,252,126$. □

Due to the geometric modeling, non-convex polytopes have to be considered in some cases. Since this non-convexity is closely related to the kind of handled problem, a convexity test is unuseful. In such a case, the parametric vertices finding algorithm has to be extended in order to consider unions of convex polytopes $P_N^1, P_N^2, \ldots$ This extension consists in the following heuristic procedure:

- Compute all the vertices of each rational polytope P_N^i.
  For each found vertex v, v is a vertex of P_N if and only if it is a vertex of a polytope P_N^i and it does not belong to any other $P_N^j, j \neq i$, unless it is also a vertex of $P_N^j, j \neq i$.

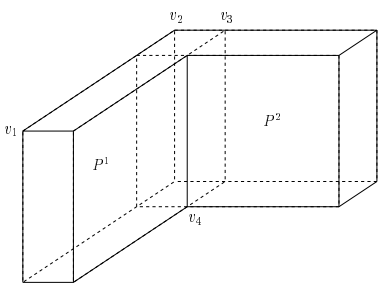

Fig. 3. A 3D illustration for the vertex finding procedure of non-convex sets.

- Compute all the vertices of each intersection $P_N^i \cap P_N^j, j \neq i$.
  For each found vertex v, v is a vertex of P_N if and only if it is not a vertex of any P_N^i.

Example: A simple 3-dimensional example $P = P^1 \cup P^2$ is shown on figure 3 :

- v_1 is a vertex of P^1 and does not belong to P^2. Hence, v_1 is a vertex of P.
- v_2 is a vertex of P^1 and is also a vertex of P^2. Hence, v_2 is a vertex of P.
- v_3 is a vertex of P^1 and belongs to P^2 without being one of its vertices. Hence, v_3 is not a vertex of P.
- v_4 is a vertex of $P^1 \cap P^2$ and is not a vertex, neither of P^1 nor of P^2. Hence, v_4 is a vertex of P.

□

Ehrhart polynomials can also be used to count integer points in non-parametric polytopes and thus can be helpful for enumerations of large domains, since it strongly reduces the cost of such exhaustive computations. The way is to introduce an artificial parameter n, to compute the corresponding Ehrhart polynomial and then to evaluate this polynomial with the value of n corresponding to the initial problem. Let us consider an example:

Example: Let the non-parametric polytope P be defined by $P = \{(i,j,k) \in \mathcal{Z}^3 \mid 1 \leq i \leq 10^6, 3 \leq j \leq i, j \leq k \leq 5\}$. An artificial parameter n is introduced, replacing 10^6, and yielding the following parametric polytope definition: $P_n = \{(i,j,k) \in \mathcal{Z}^3 \mid 1 \leq i \leq n, 3 \leq j \leq i, j \leq k \leq 5\}$. Our method computes the Ehrhart polynomial of P_n: $j_n = 6n - 16$. Hence, by computing j_{10^6}, it is obtained the number of integer points in the initial non-parametric polytope $P = P_{10^6}$, $j_{10^6} = 5,999,984$. □

Observe that these described calculations are independent of the size parameters of the considered polytope. This method is now applied in the three following sections to exact symbolic array dataflow analysis, to estimation of execution time and to the determination of the maximum parallelism of given loop nests.

5. An application of parametric analysis: Exact symbolic array dataflow analysis

Most of the proposed dependence tests included in vectorization and parallelization methods [6] are not exact, or not able to consider only true dependences, i.e. they also consider dependences where the value supposed to be transmitted is replaced before being used.

In the context of nested loops analysis, some methods have been proposed to compute for every array element read during an iteration, the very iteration which produced it. These methods are commonly called *array dataflow analyses* [7], [21], [22]. However, the used tools yield complicated methods, and do not handle symbolic answers when analyzing a given nested loop.

Such an analysis also allows to achieve other important goals such as conversion into single-assignment form or linear recurrence equation, array expansion, dead-code elimination, ...

Our method consists first in computing the polytope P_{I_0} characterizing iterations I executed before a given one I_0, according to the lexicographic order: $P_{I_0} = \{I \mid I \triangleright I_0\}$. Then, this polytope is reduced to another polytope P'_{I_0} characterizing iterations whose operations consist in updating the array element read during I_0. Finally, the answer is given by determining the iteration I_{max} which is lexicographically maximum in this reduced polytope, i.e. such that: $\forall\ I \in P'_{I_0}, I \triangleright I_{max} \triangleright I_0$.

Since the subject of this paper is not to describe the complete method on array dataflow analysis, we just want to show here the opportunity of our parametric tools in this area. Hence, a simple example is presented in the following.

Consider the loop nest:

```
for i := 1 to n do
  for j := i to n do
    a(i+j) := a(i-j+n) + 1
```

The question is, given an iteration $I_0 = (i_0, j_0)$, what is the iteration $I_{max} = (i_{max}, j_{max})$ computing the value of $a(i_0 - j_0 + n) = a(i_{max} + j_{max})$ read by I_0.

The first step is to compute the iterations $I = (i, j)$ that are executed before I_0. Since the execution order is lexicographic, these iterations are first characterized as a union of disjoint polytopes, each associated to a particular subset of iterations preceding I_0:

- iterations I characterized by $i < i_0$,
- iterations I characterized by $i = i_0$ and $j < j_0$

It yields:

$$\begin{aligned} P_{I_0} \;=\; & \{1 \le i \le i_0 - 1, i \le j \le n, i_0 \le n, \\ & \quad i_0 \le j_0 \le n\} \\ \cup\; & \{1 \le i_0 \le n, i \le j \le j_0 - 1, i = i_0, j_0 \le n\} \end{aligned}$$

Reducing this definition to the union of polytopes P'_{I_0} characterizing iterations updating $a(i_0 - j_0 + n)$ yields:

$$P'_{I_0} = P'_{I_0,1} \cup P'_{I_0,2}$$

where

$$\begin{aligned} P'_{I_0,1} \;=\; & \{1 \le i \le i_0 - 1, i \le j \le n, \\ & \quad i_0 \le n, i_0 \le j_0 \le n, i + j = i_0 - j_0 + n\} \end{aligned}$$

$$\begin{aligned} P'_{I_0,2} \;=\; & \{1 \le i_0 \le n, i \le j \le j_0 - 1, i = i_0, \\ & \quad j_0 \le n, i + j = i_0 - j_0 + n\} \end{aligned}$$

On $P'_{I_0,2}$, since the value of i is fixed to $i = i_0$, we have $i_{max} = i_0$ and j_{max} is deduced from the constraint $i + j = i_0 - j_0 + n$, i.e. $j_{max} = n - j_0$. By computing the parametric vertices of $P'_{I_0,2}$, the resulting validity domain gives the range of values of i_0 and j_0 for which $i_{max} = i_0$ and $j_{max} = n - j_0$ occurs. It yields the following intermediate result:

If $j_0 \ge \frac{n+1}{2}$ and $i_0 + j_0 \le n$, the array element read during iteration (i_0, j_0) is produced during iteration $(i_0, n - j_0)$.

On $P'_{I_0,1}$, we use our parametric vertices finding algorithm by considering the index i as a parameter. The upper bounds concerning index i given by the validity domains are then the maximum values i_{max}. The algorithm computes the following validity domain:

$$\{1 \le i \le i_0 - 1, 2i \le i_0 - j_0 + n, i_0 \le j_0 \le n\}$$

Hence, the maximum value taken by i can be either $i_{max} = i_0 - 1$ (yielding $j_{max} = n - j_0 + 1$) or $i_{max} = \lfloor \frac{i_0 - j_0 + n}{2} \rfloor$ (yielding $j_{max} = i_0 - j_0 + n - \lfloor \frac{i_0 - j_0 + n}{2} \rfloor$).

First, we add the constraint $i = i_0 - 1$ to $P'_{I_0,1}$ and compute the parametric vertices to get the validity domain for which this value of i_{max} occurs. It yields the following validity domain:

$$\{2 \leq i_0 \leq j_0, i_0 + j_0 \leq n + 2\}$$

In the same way, we now add the constraint $i_{max} = \frac{i_0 - j_0 + n}{2}$ for the case $(i_0 - j_0 + n)\ mod\ 2 = 0$ to $P'_{I_0,1}$. It yields:

$$\{i_0 \leq j_0 \leq n, i_0 + j_0 \geq n + 2\}$$

And finally, we add the constraint $i_{max} = \frac{i_0 - j_0 + n - 1}{2}$ for the case $(i_0 - j_0 + n)\ mod\ 2 = 1$ to $P'_{I_0,1}$. It yields:

$$\{i_0 \leq j_0 \leq n, i_0 + j_0 \geq n + 1, i_0 + n \geq j_0 + 3\}$$

We conclude from these results that given an iteration $I_0 = (i_0, j_0)$, the array element $a(i_0 - j_0 + n)$ read during I_0 is lastly updated at iteration $I_{max} = (i_{max}, j_{max})$ such that:

if $j_0 \geq \frac{n+1}{2}$ and $i_0 + j_0 \leq n$

$I_{max} = (i_0, n - j_0)$

else if $2 \leq i_0 \leq j_0$ and $i_0 + j_0 \leq n + 2$

$I_{max} = (i_0 - 1, n - j_0 + 1)$

else if $i_0 \leq j_0 \leq n$ and $i_0 + j_0 \geq n + 2$ and $(i_0 - j_0 + n)\ mod\ 2 = 0$

$I_{max} = (\frac{i_0 - j_0 + n}{2}, \frac{i_0 - j_0 + n}{2})$

else if $i_0 \leq j_0 \leq n$ and $i_0 + j_0 \geq n + 1$ and $i_0 + n \geq j_0 + 3$ and $(i_0 - j_0 + n)\ mod\ 2 = 1$

$I_{max} = (\frac{i_0 - j_0 + n - 1}{2}, \frac{i_0 - j_0 + n + 1}{2})$

else *undefined* (the array element is not updated in the loop)

This final result allows to deduce four kinds of dependence vectors associated to the reading of $a(i_0 - j_0 + n)$: $(0, 2j_0 - n)$, $(1, 2j_0 - n - 1)$, $(\frac{i_0 + j_0 - n}{2}, \frac{-i_0 + 3j_0 - n}{2})$ and $(\frac{i_0 + j_0 - n + 1}{2}, \frac{-i_0 + 3j_0 - n - 1}{2})$. The four disjoint iteration subsets which are ends of these vectors are not always convex. Origins of these vectors are overlapped iteration subsets, since some iterations are origins for several kinds of dependences.

6. An application of parametric analysis: Estimation of nested loop execution time

Let us consider the following loop nest:

```
for i := 0 to n do
  for j := 0 to i + m/2 do
    for k := 0 to i - n + p do
      Statement(s)
```

We want to compute the number of flops in order to evaluate the execution time of this code segment, as in [23]. This loop nest is modeled by the convex polytope $P_N = \{z \in \mathcal{Z}^3 \mid A \cdot z \leq BN + C\}$ where

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} \quad N = \begin{pmatrix} n \\ m \\ p \end{pmatrix}$$

$$C = 0$$

The parametric vertices finding algorithm computes the following vertices for the so-defined polytope P_N:

- (1) if $n > p$: $(n - p, \frac{2n+m-2p}{2}, 0)$, $(n - p, 0, 0)$, $(n, \frac{2n+m}{2}, p)$, $(n, \frac{2n+m}{2}, 0)$, $(n, 0, p)$, $(n, 0, 0)$.
- (2) if $n = p$: $(0, \frac{m}{2}, 0)$, $(n, 0, p)$, $(0, 0, 0)$, $(n, \frac{2n+m}{2}, 0)$, $(n, 0, 0)$, $(n, \frac{2n+m}{2}, p)$.
- (3) if $n < p$: $(0, \frac{m}{2}, -n + p)$, $(0, \frac{m}{2}, 0)$, $(0, 0, -n + p)$, $(0, 0, 0)$, $(n, \frac{2n+m}{2}, p)$, $(n, \frac{2n+m}{2}, 0)$, $(n, 0, p)$, $(n, 0, 0)$.

As an example, we now compute the Ehrhart polynomial on the first definition domain.

The denominator of P_N is 2. We count $((k + 1)d)^q = 512$ initial values of $j(n, m, p)$. The lowest initial value of n is 9 since we consider the first definition domain $(n > p)$ and since for any initial value of n, 8 initial values of p have to be considered. For example, with $n = 9$ we get the following initial countings:

m \ p	1	2	3	4	5	6	7	8
1	29	56	90	130	175	224	276	330
2	32	62	100	145	196	252	312	375
3	32	62	100	145	196	252	312	375
4	35	68	110	160	217	280	348	420
5	35	68	110	160	217	280	348	420
6	38	74	120	175	238	308	384	465
7	38	74	120	175	238	308	384	465
8	41	80	130	190	259	336	420	510

Each line of such arrays allows to resolve a system of equations to determine $j_{x,y,p}$, for all $x \in [9..16]$ and $y \in [1..8]$. For example, the first system dedicated to the determination of $j_{9,1,p} = [c_1, c_2]_p \, n_3^3 + [c_3, c_4]_p \, n_3^2 + [c_5, c_6]_p \, n_3 + [c_7, c_8]_p$ is

$$\begin{cases} c_1 + c_3 + c_5 + c_7 = 29 \\ 8c_2 + 4c_4 + 2c_6 + c_8 = 56 \\ 27c_1 + 9c_3 + 3c_5 + c_7 = 90 \\ 64c_2 + 16c_4 + 4c_6 + c_8 = 130 \\ 125c_1 + 25c_3 + 5c_5 + c_7 = 175 \\ 216c_2 + 36c_4 + 6c_6 + c_8 = 224 \\ 343c_1 + 49c_3 + 7c_5 + c_7 = 276 \\ 512c_2 + 64c_4 + 8c_6 + c_8 = 330 \end{cases}$$

resulting in $j_{9,1,p} = -\frac{1}{6}p^3 + \frac{9}{2}p^2 + \frac{44}{3}p + 10$.

All these last polynomials allows to resolve systems to determine $j_{x,m,p}$, for all $x \in [9..16]$. For example, the first system dedicated to the determination of $j_{9,m,p} = [c_1, c_2]_m \, m^3 + [c_3, c_4]_m \, m^2 + [c_5, c_6]_m \, m + [c_7, c_8]_m$ is:

$$\begin{cases} c_1 + c_3 + c_5 + c_7 = -\frac{1}{6}p^3 + \frac{9}{2}p^2 + \frac{44}{3}n_3 + 10 \\ 8c_2 + 4c_4 + 2c_6 + c_8 = -\frac{1}{6}p^3 + 5p^2 + \frac{97}{6}p + 11 \\ 27c_1 + 9c_3 + 3c_5 + c_7 = -\frac{1}{6}p^3 + 5p^2 + \frac{97}{6}p + 11 \\ 64c_2 + 16c_4 + 4c_6 + c_8 = -\frac{1}{6}p^3 + \frac{11}{2}p^2 + \frac{53}{3}p + 12 \\ 125c_1 + 25c_3 + 5c_5 + c_7 = -\frac{1}{6}p^3 + \frac{11}{2}p^2 + \frac{53}{3}p + 12 \\ 216c_2 + 36c_4 + 6c_6 + c_8 = -\frac{1}{6}p^3 + 6p^2 + \frac{115}{6}p + 13 \\ 343c_1 + 49c_3 + 7c_5 + c_7 = -\frac{1}{6}p^3 + 6p^2 + \frac{115}{6}p + 13 \\ 512c_2 + 64c_4 + 8c_6 + c_8 = -\frac{1}{6}p^3 + \frac{13}{2}p^2 + \frac{62}{3}p + 14 \end{cases}$$

In this case, the following result is obtained:

$$j_{9,m,p} = \tfrac{1}{4}mp^2 + \tfrac{3}{4}mp + \tfrac{1}{2}m - \tfrac{1}{6}p^3 + \left[\tfrac{17}{4}, \tfrac{9}{2}\right]_m p^2 + \left[\tfrac{167}{12}, \tfrac{44}{3}\right]_m p + \left[\tfrac{19}{2}, 10\right]_m$$

Finally, the following result is obtained for $j_{n,m,p}$, giving the number of flops executed by the loop nest when $n > p$:

$$j_{n,m,p} = \tfrac{1}{2}np^2 + \tfrac{3}{2}np + n + \tfrac{1}{4}mp^2 + \tfrac{3}{4}mp + \tfrac{1}{2}m - \tfrac{1}{6}p^3 + \left[-\tfrac{1}{4}, 0\right]_m p^2 + \left[\tfrac{5}{12}, \tfrac{7}{6}\right]_m p + \left[\tfrac{1}{2}, 1\right]_m$$

Notice that this number has eventually to be multiplied by the number of statements that are inside the analyzed loop nest.

7. An application of parametric analysis: the maximum parallelism

The knowledge of the maximum parallelism is essential in optimizing nested-loop parallelization processes, since the minimum number of processors needed to exploit all the inherent parallelism is thus known. Such a utilisation of this information is shown in [24], [25], [26] in the context of optimal mappings of systolic algorithms. Even if the minimum number of processors can reasonably not be reached during the mapping process, this value is useful to evaluate the efficiency of one another parallelization method, since fixed-sized processor array mapping may in some worse cases result in completely sequential programs.

Consider the following loop nest:

```
for i := 0 to n do
  for j := 0 to i do
    for k := 0 to i-j do
      Statement(s)
```

This loop nest is modeled by the one-parameter polyhedral iteration space defined by:

$$P_n = \{ \begin{pmatrix} i \\ j \\ k \end{pmatrix} \in \mathcal{Z}^3 \mid \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} \leq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} n \}$$

Assume that the dependence analysis of this loop nest yields an optimal linear schedule defined by $t(i,j,k) = i + j + k$: any iteration (i,j,k) is computed at instant $i + j + k$. We use our method to determine the parametric number of simultaneous calculations $par(n, t_0)$ occurring at any instant t_0, in order to determine the parametric maximum parallelism $maxpar(n) = MAX_{t_0}\{par(n, t_0)\}$, i.e. the maximum number of simultaneous calculations.

A multi-dimensional linear schedule could also have been considered, since such schedules are simply to be modeled by lower dimensional hyperplanes.

From a geometrical point of view, all the points (i,j,k) of P_n resulting in the same value of $t_0 =$

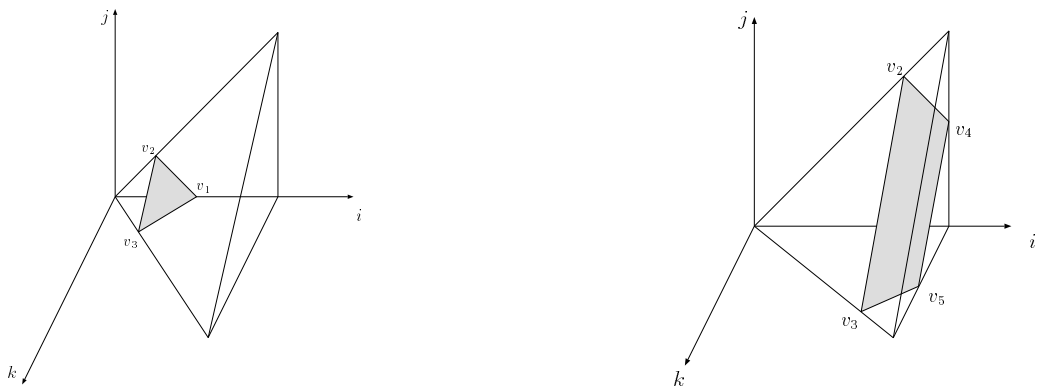

Fig. 4. T_{n,t_0} in the first and second validity domains

$t(i,j,k)$, belong to the two-parameters rational 2-polytope T_{n,t_0} defined by $T_{n,t_0} = P_n \cap \{(i,j,k) \in \mathcal{Z}^3 | i+j+k = t_0\}$. The number of points in T_{n,t_0} is $par(n,t_0)$.

Giving this last parametric definition of T_{n,t_0} as an input to the parametric vertices finding algorithm, it outputs the following results as shown on figure 4:

- if $0 \leq t_0 \leq n$, the parametric vertices are $v_1 = (t_0,0,0)$, $v_2 = \left(\frac{t_0}{2}, \frac{t_0}{2}, 0\right)$ and $v_3 = \left(\frac{t_0}{2}, 0, \frac{t_0}{2}\right)$.
- if $n \leq t_0 \leq 2n$, the parametric vertices are $v_2 = \left(\frac{t_0}{2}, \frac{t_0}{2}, 0\right)$, $v_3 = \left(\frac{t_0}{2}, 0, \frac{t_0}{2}\right)$, $v_4 = (n, t_0 - n, 0)$ and $v_5 = (n, 0, t-n)$.
- otherwise, T_{n,t_0} does not exist and $par(n,t_0) = 0$.

Hence, $par(n,t_0)$ is defined on two validity domains as pseudo-polynomials in n and t_0, of degree 2, and of pseudo-period 2. According to theorem 4 and by looking at $T_{1,0}$, it is deduced that all the coefficients of n^q, $q \leq 2$, are constant: $par(n,t_0) = c_1 n^2 + c_2 n + c_3$.
(i) We determine c_1, c_2, c_3 by solving the following system of equations on the first validity domain:

$$\begin{cases} 36c_1 + 6c_2 + c_3 &= par(6,t_0) \\ 49c_1 + 7c_2 + c_3 &= par(7,t_0) \\ 64c_1 + 8c_2 + c_3 &= par(8,t_0) \end{cases}$$

Solving this system yields the equations for $par(6,t_0)$, $par(7,t_0)$ and $par(8,t_0)$. According to theorem 4 and by looking at $T_{1,1}$, it is deduced that only the coefficient of t_0^2 is constant. For example, $par(6,t_0) = c_1 t_0^2 + [c_2,c_3]t_0 + [c_4,c_5]$, such that:

$$\begin{cases} c_5 &= par(6,0) = 1 \\ c_1 + c_2 + c_4 &= par(6,1) = 1 \\ 4c_1 + 2c_3 + c_5 &= par(6,2) = 3 \\ 9c_1 + 3c_2 + c_4 &= par(6,3) = 3 \\ 16c_1 + 4c_3 + c_5 &= par(6,4) = 6 \end{cases}$$

where the values of $par(6,0), \ldots, par(6,4)$ are given by some initial countings.
$\Rightarrow$ $par(6,t_0) = \frac{1}{8}t_0^2 + [\frac{1}{2}, \frac{3}{4}]t_0 + [\frac{3}{8}, 1]$. In the same way, we determine that the other pseudo-polynomials are the same, i.e $\forall\ n \in [6..8]$, $par(n,t_0) = \frac{1}{8}t_0^2 + [\frac{1}{2}, \frac{3}{4}]t_0 + [\frac{3}{8}, 1]$. Solving the first system gives the parallelism rate of any instant t_0 on the first validity domain:
$\forall\ 0 \leq t_0 \leq n$, $par(n,t_0) = \frac{1}{8}t_0^2 + \left[\frac{1}{2}, \frac{3}{4}\right] t_0 + \left[\frac{3}{8}, 1\right]$
Observe that the parallelism rate does not depend on n in this validity domain.
(ii) We now consider the second validity domain.

$$\begin{cases} 36c_1 + 6c_2 + c_3 &= par(6,t_0) \\ 49c_1 + 7c_2 + c_3 &= par(7,t_0) \\ 64c_1 + 8c_2 + c_3 &= par(8,t_0) \end{cases}$$

Solving this system yields the equations for $par(6,t_0)$, $par(7,t_0)$ and $par(8,t_0)$. For example, $par(6,t_0) = c_1 t_0^2 + [c_2,c_3]t_0 + [c_4,c_5]$, such that:

$$\begin{cases} 36c_1 + 6c_3 + c_5 &= par(6,6) &= 10 \\ 49c_1 + 7c_2 + c_4 &= par(6,7) &= 9 \\ 64c_1 + 8c_3 + c_5 &= par(6,8) &= 12 \\ 81c_1 + 9c_2 + c_4 &= par(6,9) &= 9 \\ 100c_1 + 10c_3 + c_5 &= par(6,10) &= 11 \end{cases}$$

$\Rightarrow par(6,t_0) = -\frac{3}{8}t_0^2 + [6,\frac{25}{4}]t_0 + [-\frac{117}{8},-14]$. In the same way, we determine the other pseudo-polynomials:

$$\begin{aligned} par(7,t_0) &= -\tfrac{3}{8}t_0^2 + [7,\tfrac{29}{4}]t_0 + [-\tfrac{165}{8},-20] \\ par(8,t_0) &= -\tfrac{3}{8}t_0^2 + [8,\tfrac{33}{4}]t_0 + [-\tfrac{221}{8},-27] \end{aligned}$$

Solving the first system gives the parallelism rate of any instant t_0 on the second validity domain:

$$\forall\ n \le t_0 \le 2n, par(n,t_0) = nt_0 - \tfrac{1}{2}n^2 + \tfrac{1}{2}n - \tfrac{3}{8}t_0^2 + \left[0,\tfrac{1}{4}\right]_{t_0} t_0 + \left[\tfrac{3}{8},1\right]_{t_0}$$

(iii) From these results, the maximum parallelism is deduced:

- On the first validity domain ($0 \le t_0 \le n$), $par(n,t_0)$ is maximum when t_0 is maximum, i.e. when $t_0 = n$. Hence, $maxpar(n) = \frac{1}{8}n^2 + \left[\frac{1}{2},\frac{3}{4}\right] n + \left[\frac{3}{8},1\right]$.
- On the second validity domain ($n \le t_0 \le 2n$), we take the derivative of $par(n,t_0)$ with respect to t_0: $par'(n,t_0) = n - \frac{3}{4}t_0 + \left[0,\frac{1}{4}\right]_{t_0}$.

$$\begin{aligned} & par'(n,t_0) \le 0 \\ \Leftrightarrow\ & n - \tfrac{3}{4}t_0 + \left[0,\tfrac{1}{4}\right]_{t_0} \le 0 \\ \Leftrightarrow\ & t_0 \ge \tfrac{4n-[0,1]}{3} \end{aligned}$$

  Hence, $par(n,t_0)$ increases from $t_0 = n$ to $t_0 = \frac{4n-[0,1]}{3}$, and then decreases until $t_0 = 2n$: it is maximum for $t_0 = t_{max} = \frac{4n-[0,1]}{3}$. Since t_{max} is not always integral, a case study is necessary to get the exact value of $maxpar(n) = par(n,t_{max})$, checking the possible values of $\lfloor t_{max} \rfloor$ and $\lceil t_{max} \rceil$. It yields the following answer: $maxpar(n) = \frac{1}{6}n^2 + \frac{5}{6}n - \left[1,\frac{2}{3},1\right]$.
- In conclusion, the maximum parallelism of the loop nest is the maximum of both previously found maxima:

  $\forall\ n \ge 0,\ maxpar(n) = \frac{1}{6}n^2 + \frac{5}{6}n - \left[1,\frac{2}{3},1\right]$

8. Conclusion

We have presented methods for computing the parametric vertices and their associated domains of a polytope defined by parametric linear constraints, and for computing the exact parametric number of integer points, i.e. the Ehrhart polynomial, of a convex polytope or a union of convex polytopes. These methods include an original algorithm to find parametric vertices and an extension of Ehrhart's results to any number of parameters.

The procedures are entirely automatic, from a parametric polytope definition as a system of linear constraints to the parametric vertices and the Ehrhart polynomials. It is implemented at this time. It uses the *polyhedral library* [27] and is available at the following address: http://icps.u-strasbg.fr/Ehrhart/program/program.html.

This is the first method to be proposed that enumerates both integer and rational polytopes using a single coherent algorithm. Moreover, the output in the form of Ehrhart polynomials is simple and gives exact parametric expressions for the number of integer points. Other work dealing with this same problem is presented below.

M. Haghighat and C. Polychronopoulos present in [5], [28] a method for volume computation. The form of their answers is different because of the *min* and *max* expressions they introduce. Their results tend to be more complicated.

Ferrante, Sarkar and Thrash in [2] present methods for computing the number of distinct memory locations and cache lines accessed by a loop nest. These methods cannot handle unions of polytopes, do not compute symbolic answers, often compute only an approximation and use expensive methods to handle the cache lines touched by a set of references.

The method described by Pugh in [1] consists in a set of techniques, dedicated to simplifying the initial set of constraints. Applying all these tools would be expensive for important sets of constraints with rational coefficients. Moreover, their application is often intuitive: it consists in some cases in adding auxiliary variables, and requires the use of standard formulae for sums of powers (the author expects it will be sufficient to hard code the formulae for powers up to 10). All these facts clearly show some limits in the general use of these techniques.

Tawbi in [4], [23] first uses a polyhedral splitting technique to characterize the validity domains, and an elimination of the variables in a predetermined order. However, her method does not systematically compute exact answers like ours.

We have illustrated this paper with three examples from possible applications. There are a lot of other applications in the analysis and transforma-

tion of loop nests, some of which can be found in works cited hereabove or in [29], [30]. Other applications may also be found in other fields as sequential program analysis, combinatorics and optimization.

For future works, we believe that many other computations, which are useful in automatic parallelization, can be achieved through our parametric tools. Hence, we are currently studying the construction of some other opportune models and methods.

Acknowledgements

We are grateful to Doran K. Wilde for his useful comments and suggestions. We also would like to thank the referees for their helpful comments.

References

1. W. Pugh. Counting Solutions to Presburger Formulas: How and Why. *Proc. of the 1994 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 1994.
2. J. Ferrante, V. Sarkar and W. Thrash. On estimating and enhancing cache effectiveness. *Advances in Languages and Compilers for Parallel Processing*, pages 328-343. The MIT Press, 1991.
3. H. S. Stone and D. Thiebaut. Footprints in the cache. *Proc. ACM SIGMETRICS 1986*, pp. 4-8, May 1986.
4. N. Tawbi and P. Feautrier. Processor allocation and loop scheduling on multiprocessor computers. *Proc. of the 1992 International Conference on Supercomputing*, pages 63-71, July 1992.
5. M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. *U. Banerjee et al., editor, Languages and Compilers for Parallel Computing*. Springer-Verlag, August 1993. LNCS 768, *Proc. of the 6th annual worshop on Programming Languages and Compilers for Parallel Computing*.
6. U. Banerjee. *Dependence Analysis for Supercomputing*. Kluwer Academic Publishers, Boston/Dordrecht/London, 1988.
7. P. Feautrier. Dataflow analysis of scalar and array references. *Int. J. of Parallel Programming*, 20(1):23-53, Feb. 1991.
8. P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. North-Holland, Amsterdam, 1987.
9. A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 1986.
10. M.S. Bazaraa, H.D. Sherali and C.M. Shetty. *Nonlinear Programming, Theory and Algorithms, second edition*. Wiley, New York, 1993.
11. R. P. Stanley. *Combinatorics and Commutative Algebra*. Birkhäuser, Boston, 1983.
12. R. P. Stanley. *Enumerative Combinatorics Volume I*. Wadsworth & Brooks/Cole, Monterey, California, 1986.
13. T. H. Mattheiss and D. Rubin. A survey and comparison of methods for finding all vertices of convex polyhedral sets. *Mathematics of Operations Research*, 5(2):167-185, 1980.
14. E. Ehrhart. Sur un problème de géométrie diophantienne linéaire I. *J. Reine Angew. Math.*, 226:1-29, 1967.
15. E. Ehrhart. *Polynômes arithmétiques et Méthode des Polyèdres en Combinatoire*. International Series of Numerical Mathematics, vol.35, Birkhäuser Verlag, Basel/Stuttgart, 1977.
16. V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. *Int. J. of Parall. Progr* 25(6), Dec. 1997. Research Report ICPS 96-09, 1996. `http://icps.u-strasbg.fr/pub-95/pub-96-09.ps.gz`
17. E. Ehrhart. Sur les polyèdres rationnels homothétiques à n dimensions. *C.R. Acad. Sci. Paris*, 254:616-618, 1962.
18. E. Ehrhart. Sur un problème de géométrie diophantienne linéaire II. *J. Reine Angew. Math.*, 227:25-49, 1967.
19. I. G. Macdonald. The volume of a lattice polyhedron. *Proc. Camb. Phil. Soc.*, 59:719-726, 1963.
20. I. G. Macdonald. Polynomials associated with finite cell-complexes. *J. London Math. Soc.*, 4(2):181-192, 1971.
21. D. E. Maydan, S. P. Amarasingue, M. S. Lam. Array dataflow analysis and its use in array privatization. In *Proc. of ACM Conf. on Principles of Programming Languages*, pages 2-15, Jan. 1993.
22. E. Duesterwald, R. Gupta, M.-L. Soffa. A practical data flow framework for array reference analysis and its use in optimization. In *ACM SIGPLAN'93 Conf. on Prog. Lang. Design and Implementation*, pages 68-77, June 1993.
23. N. Tawbi. Estimation of nested loop execution time by integer arithmetics in convex polyhedra. *Proc. of the 1994 International Parallel Processing Symposium*, April 1994.
24. Ph. Clauss, C. Mongenet and G.-R. Perrin. Calculus of space-optimal mappings of systolic algorithms on processor arrays, *Journal of VLSI Signal Processing*, 4, pages 27-36, 1992.
25. Ph. Clauss. An efficient allocation strategy for mapping affine recurrences into space and time optimal regular processor arrays. In C. Jesshope, V. Jossifov and W. Wilhelmi editors, Int. Conf. PARCELLA'94, pages 257-266, Potsdam, Germany, Sept. 1994, Akademie Verlag. `http://icps.u-strasbg.fr/pub-94/pub-94-15.ps.Z`
26. Ph. Clauss and G.-R. Perrin. Optimal mapping of systolic algorithms by regular instruction shifts. Int. Conf. on Application-Specific Array Processors, ASAP'94, pages 224-235, San Francisco, California, IEEE Computer Society Press. `http://icps.u-strasbg.fr/pub-94/pub-94-08.ps.Z`
27. D. K. Wilde. A library for doing polyhedral operations. *Master's thesis*, Oregon State University,

Corvallis, Oregon, december 1993. Also published as IRISA technical report PI 785, Rennes, France, december 1993.
28. M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. Technical Report 1317, CSRD, Univ. of Illinois, August 1993.
29. Ph. Clauss. The volume of a lattice polyhedron to enumerate processors and parallelism. Research Report ICPS 95-11, 1995. `http://icps.u-strasbg.fr/pub-95/pub-95-11.ps.gz`
30. Ph. Clauss. Counting Solutions to Linear and Nonlinear Constraints through Ehrhart polynomials: Applications to Analyze and Transform Scientific Programs. *10th ACM Int. Conf. on Supercomputing*, Philadelphia, May 1996. Also available as Research Report ICPS 96-03. `http://icps.u-strasbg.fr/pub-96/pub-96-03.ps.gz`

Philippe Clauss is an assistant professor of computer science at the University Louis Pasteur of Strasbourg, France. His research interests include automatic parallelization of programs, integer and rational geometry and non-linear mapping functions of calculations and data. He received the PhD (1990) in Computer Science from the University of Franche-Comté.

Vincent Loechner is a PhD student at the ICPS laboratory of the University Louis Pasteur, Strasbourg, France. His research interests include automatic parallelization of programs, communication minimization and integer and rational geometry. He is managing the development of the OPERA research project. He has just received the PhD (1997) in Computer Science from the University Louis Pasteur of Strasbourg.