Counting Solutions to Linear and Nonlinear Constraints Through Ehrhart Polynomials: Applications to Analyze and Transform Scientific Programs

Conference Paper · January 1996

DOI: 10.1145/237578.237617 · Source: DBLP

Philippe Clauss
INRIA, CNRS, University of Strasbourg

Counting Solutions to Linear and Nonlinear Constraints through Ehrhart polynomials: Applications to Analyze and Transform Scientific Programs

Philippe CLAUSS
ICPS, Université Louis Pasteur, Strasbourg,
Pôle API, Bd Sébastien Brant,
67400 Illkirch France
clauss@icps.u-strasbg.fr
http://icps.u-strasbg.fr/people/philippe.html

Abstract

In order to produce efficient parallel programs, optimizing compilers need to include an analysis of the initial sequential code. When analyzing loops with affine loop bounds, many computations are relevant to the same general problem: counting the number of integer solutions of selected free variables in a set of linear and/or nonlinear parameterized constraints. For example, computing the number of flops executed by a loop, of memory locations touched by a loop, of cache lines touched by a loop, or of array elements that need to be transmitted from a processor to another during the execution of a loop, is useful to determine if a loop is load balanced, evaluate message traffic and allocate message buffers.

The objective of the presented method is to evaluate symbolically, in terms of symbolic constants (the size parameters), this number of integer solutions. By modeling the considered counting problem as a union of rational convex polytopes, the number of included integer points is expressed by particular polynomials where the free variables are the size parameters, commonly called *Ehrhart polynomials*. The paper is illustrated by many examples dealing with parallel program optimizations.

1 Introduction

In order to produce efficient parallel programs, optimizing compilers need to include an analysis of the initial sequential code involving the computation of some determinant values. For example, computing the number of flops executed by a loop, of memory locations touched by a loop, of cache lines touched by a loop, or of array elements that need to be transmitted from a processor to another during the execution of a loop, is useful to determine if a loop is load balanced, evaluate message traffic and allocate message buffers. Such computations are relevant to the same general problem: counting the number of integer solutions of selected free variables in a set of linear and/or nonlinear constraints.

This set of constraints depends on some positive integral size parameters. Any constraint C_k is of the form

$$C_k : \sum_i a_i \alpha_i \leq \sum_j b_j n_j + c$$

where the a_i's, b_i's and c are rational constants, the n_j's are the size parameters and the α_i's can be of the form

$$x_i \quad (x_i \ mod \ d_i) \quad \lfloor \tfrac{x_i}{d_i} \rfloor \quad \lceil \tfrac{x_i}{d_i} \rceil \quad (d_i | x_i)$$

where d_i is an integral constant, $(d_i|x_i)$ means that d_i evenly divides x_i and x_i is any rational linear combination of the free variables. When the form of any α_i yields a nonlinear constraint, this can easily be transformed into a linear one through some elementary transformations, as those presented by W. Pugh in [16]:

• If a term $\lfloor \frac{x_i}{d_i} \rfloor$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i z \leq x_i < d_i(z+1), C'_k\}$ where C'_k is C_k with $\lfloor \frac{x_i}{d_i} \rfloor$ replaced with z.
• If a term $\lceil \frac{x_i}{d_i} \rceil$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i(z-1) < x_i \leq d_i z, C'_k\}$ where C'_k is C_k with $\lceil \frac{x_i}{d_i} \rceil$ replaced with z.
• If a term $(x_i \ mod \ d_i)$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i z \leq x_i < d_i(z+1), C'_k\}$ where C'_k is C_k with $(x_i \ mod \ d_i)$ replaced with $x_i - z$.
• A constraint $(d_i|x_i)$ is equivalent to $(x_i \ mod \ d_i = 0)$ and the precedent rule can be applied.
• If a term of the form $\lfloor \frac{U}{u} \rfloor$ appears in the right side of the constraint C_k, since rational constraints are considered by our method, the floor can be avoided and the whole constraint is multiplied by u.

The objective is to evaluate, in terms of symbolic constants (the size parameters), the number of integer solutions of a combination of linear and nonlinear constraints C_k. This number is expressed by polynomials where the free variables are the size parameters.

As said before, there are a lot of applications of such a counting, in the area of analysis and transformation of scientific programs expressed as loops with affine loop bounds. The polynomials may count the flops executed by a loop, the memory locations touched by a loop, the cache lines touched by a loop, the array elements that need to be transmitted from a processor to another during the execution of a loop, the maximum parallelism induced by a loop from a given time-schedule, the number of processors needed from a given mapping function, the amount of communications from a given loop and data partitionning, the number of

Tenth ACM International Conference on Supercomputing, May 1996, Philadelphia, Pennsylvania.

cache misses from a given loop and data partitioning (size of the footprints [18]).

Such informations can be determinant to estimate the execution time of a code segment, compare the memory bandwidth requirements vs. the flop counts of a code segment [16], determine which loops will flush the cache, and then calculate the cache miss rate [9], determine whether a parallel loop is load balanced, i.e., does each iteration perform the same number of flops [22], given an unbalanced loop, assign different number of iterations to each processor so that each processor gets the same total number of flops (balanced chunk-scheduling, [11]), quantify message traffic, allocate space for message buffers, exploit maximum parallelism, optimize the number of needed processors, optimize the loop and data partioning.

A system of constraints defines a combination of convex polytopes in a lattice of rational points. Hence, the initial problem is equivalent to counting the number of lattice points included in polytopes.

This number can be expressed as a symbolic sum, which is not generally an easy transformation. Some symbolic mathematical packages, such as Maple or Mathematica, compute such symbolic sums, but they assume that the lower bound is never greater than the upper bound. Therefore, their answer is wrong if this assumption is violated. For example, Mathematica computes that

$$\sum_{i=1}^{n}\sum_{j=i}^{m} 1 = \frac{n(2m-n+1)}{2}$$

This answer is valid only if $n \leq m$. Another answer is necessary in the case $n > m$ which is $\frac{m(m+1)}{2}$. It comes that in the general case, the symbolic number of integer solutions of a combination of constraints must by given at domains depending on the related values of the size parameters. This fact was already pointed out by the mathematician E. Ehrhart in 1977 [8]:

Conjecture 1 (Ehrhart's conjecture) *For any significant diophantine linear system of any dimension, linearly dependent of several positive integral parameters, the symbolic number of integer solutions depending on these parameters is expressed at different domains by different pseudo-polynomials.*

A *pseudo-polynomial* is a polynomial where the constant coefficients are varying relatively to the modulos of the size parameters. A more formal definition is given in the next section. E. Ehrhart has shown that the values of these coefficients are closely related to the vertices of the polytopes. Hence, the validity domains associated with the pseudo-polynomials are those associated with the parametric values of the vertices. So, this needs to determine the vertices of parameterized polytopes. To solve this problem, the new parametric vertices finding algorithm of Loechner and Wilde, fully detailed in [13], can be used. Such an algorithm allows to get all the useful informations to generalize Ehrhart's results, and to constitute a systematic method which is entirely computable.

In order to introduce the main results of this paper, we first present the so-called *polytope model*, by giving some geometric and arithmetic definitions and some useful Ehrhart's contributions. They are dealing with the computation of the polynomials, commonly called *Ehrhart polynomials*, associated with the number of lattice points included in a particular combination of convex polytopes called *normal polyhedron*. Ehrhart's results deal with convex polytopes or unions of convex polytopes, defined by linear constraints. Finally, it is shown in section 3 how these results can apply to determine useful values in analysis and transformation of scientific programs: they are illustrated with many examples and compared with related works [9, 22, 11, 12, 21, 16].

2 The polytope model

2.1 Definitions and assumptions

We first recall some basic notions dealing with geometry of numbers [10, 17] and enumerative combinatorics [19, 20]. Then some more specific concepts, closely dedicated to the scope of the paper, are introduced.

In the following, Q denotes the set of rational numbers. A *convex polyhedron* P is a subset of Q^d that is the intersection of a finite number of closed halfspaces. A bounded convex polyhedron is called a *convex polytope*. The *affine span* of P is the smaller affine subspace of Q^d entirely containing P. The *dimension*, $\dim P$, of a polyhedron P is the dimension of its affine span, and a k-dimensional polyhedron is called a k-polyhedron for brevity.

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 Q^d, called a basis for the lattice. The set of all the integral points is called *standard lattice* 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 [10]. Hence, without loss of generality, only the standard lattice $\mathcal{Z}^d$ is considered in the following. The *enumerator* of any set S is the number of points of $S \cap \mathcal{L}^d$.

The intersection between a lattice $\mathcal{L}^d$ and the convex hull of a finite number of points $z_1, \ldots, z_{m+1}$ in Q^d is called a lattice polytope P of $\mathcal{L}^d$. It is then homeomorphic to a ball B^k. We write $k = \dim P$ and call P a k-polytope. If $z_1, \ldots, z_{m+1}$ are affinely independent points of Q^d, which means that $z_2 - z_1$, $z_3 - z_1$, $\ldots$, $z_{m+1} - z_1$ are linearly independent, then the convex hull of $z_1, \ldots, z_{m+1}$ is called a m-dimensional lattice simplex S of $\mathcal{L}^d$ with vertices $z_1, \ldots, z_{m+1}$. A lattice polytope or a lattice simplex is then defined by a set of linear inequalities of the form $\{z \in \mathcal{L}^d \mid A\cdot z \leq b\}$ where A is a $m\times d$ rational matrix and b a m-dimensional rational vector.

If $\vec{c}$ is a nonzero vector and γ is defined by

$$\gamma = MAX\{\vec{c}\cdot z \mid A\cdot z \leq b\}$$

the affine hyperplane $\{z \mid \vec{c}\cdot z = \gamma\}$ 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. 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 lattice polyhedra which may be non-convex, called *normal polyhedra*, according to the following definition. This definition corresponds to the one given by Ehrhart in [6] and [8, p. 47].

Definition 1 (normal polyhedron) *A normal polyhedron P is a closed polyhedron which admits a finite rectilinear simplicial subdivision.*

A lattice polyhedron P of $\mathcal{L}^d$ is a $\mathcal{L}^d$-polyhedron if all its vertices are in $\mathcal{L}^d$. If $\mathcal{L}^d = \mathcal{Z}^d$, P is an *integral polyhedron*. If some vertex of P do not belong to $\mathcal{L}^d$ and have rational coordinates, P is a *rational* polyhedron. Since by definition, any lattice $\mathcal{L}^d$ can be deduced from $\mathcal{Z}^d$ by a non-singular transformation G, any $\mathcal{L}^d$-polyhedron is affinely equivalent to an integral polyhedron. Hence, results given in the following and concerning integral polyhedra, can be directly extended to $\mathcal{L}^d$-polyhedra. From now on and without loss of generality, the generic term of integral polyhedra will be used.

Many scientific methods and applications are defined parametrically in order to reach any size of the considered items. Implementations of these methods yields programs including loops with parameterized affine loop bounds. When modeling such loops with convex polytopes, parameterized polytopes have to be considered. Such polytopes were defined by E. Ehrhart as *homothetic* and *bordered systems* [6, 7], since his most elaborated results handle only polytopes depending on one size parameter:

Definition 2 (Ehrhart's homothetic system) *A system is homothetic if its relations are of the form* $\sum a_i z_i = bn$, $\sum a_i x_i < bn$, $\sum a_i x_i \leq bn$, *where the* a_i*'s and* b *are given integers, the* x_i*'s are free variables and* n *is a positive integral parameters. Such a system* H_n *is significant if and only if* H_1 *defines a polyhedral domain* P_1 *(obviously convex)*[1]. *In this case,* H_n *defines the domain* P_n*, which corresponds to* P_1 *by the* n*-homothetic transformation of center* O. P_1 *is called the primitive domain of* H_n.

The properties of such homothetic polyhedra yield a main mathematical result in the following. However this definition has to be extended to *bordered polytopes* in order to handle polytopes modeling loops with *affine* loop bounds.

Definition 3 (Ehrhart's bordered system) *A system* B_n *is bordered if its relations are of the form* $\sum a_i x_i = bn + c$, $\sum a_i x_i < bn + c$, $\sum a_i x_i \leq bn + c$, *where the* a_i*'s and the* b*'s are given integers, the* c*'s are rationals or not, the* x_i*'s are free variables and* n*'s is a positive integral parameter.*

By deleting all the c*'s, an homothetic system* H_n *defining the domain* P_n *is obtained. By adding a parallel hyperplane of the same dimension* $(k-1)$ *to each facet of* P_n*, at a rational distance or not, independently of the parameter* n*, but which may depend on the facet, it is obtained the bordered polytope* P'_n *whose kernel is* P_n.

In a general way, a parameterized polyhedron is homothetic or bordered at different domains depending on the values of its size parameters. In the following, we consider any k-polyhedron P_N defined by a union of homothetic or bordered systems of the form $\{z \in \mathcal{L}^d \mid A \cdot z \leq BN + C\}$, where $N = (n_1, \ldots, n_q)$, since any equality $\sum a_i z_i = \sum b_j n_j + c$ is equivalent to the two inequalities $\sum -a_i z_i \leq -\sum b_j n_j - c$ and $\sum a_i z_i \leq \sum b_j n_j + c$.

In the following, we denote by j_N the enumerator of P_N. It will be determined by *polynomials* and *pseudo-polynomials* according to the following definitions. We extend Ehrhart's definition of periodic numbers [8] to any dimension.

Definition 4 (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* $(u_I)_N$, $I = (i_1, i_2, \ldots, i_q)$*, of size* $p_1 \times p_2 \times \ldots \times p_q$ *such that*

$$u(N) = u_I\ if\ \forall j \in [1 \ldots q],\ n_j = i_j\ mod\ p_j$$

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

Ehrhart in [8] gives the basic properties of one-dimensional periodic numbers which can be easily extended to multi-dimensional ones:

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 lowest 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 $u(N)u'(N) = (u_I u'_I)_N$.

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

Definition 5 (pseudo-polynomial) *A polynomial* $f(N)$ *is a pseudo-polynomial if some of its coefficients are periodic numbers instead of being constants. The lowest common multiple of the periods of its periodic coefficients is the pseudo-period of* $f(N)$*. So a pseudo-polynomial has two characteristics, its degree and its pseudo-period.*

The results presented in the next section are those of E. Ehrhart which are limited to homothetic or bordered polyhedra of only one parameter n. We will present general extensions of these results to the case of any number of parameters in further works. Nevertheless an example is already given in section 3, showing clearly that the extension is straightforward.

2.2 The Ehrhart polynomial on normal polyhedra

We recall in this section some results due to E. Ehrhart who made a study of the enumeration of lattice points in polyhedra, by means of polynomials in the way of Euler. All the proofs can be found in [5, 6, 7, 8]. Some aspects of Ehrhart's work were corrected, streamlined, and expanded by I. G. Macdonald [14, 15] and R. P. Stanley [19, 20].

Definition 6 (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 lowest common multiple of the denominators of its vertices.*

Theorem 1 (fundamental theorem) *The enumerator* j_n *of any homothetic or 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.*

When considering integral polyhedra, the following results can be used, which allow to determine the polynomials of the enumerator from k of its initial values.

Theorem 2 (Ehrhart polynomial) *For any integral* k*-polyhedron* P_n*, the enumerator verifies the following identities:*

$$\{(j - j_0)^{k+1}\} = 0 \quad j_n = \sum_{q=1}^{k} C_n^q (j - j_0)^q + j_0 \quad j^r \sim j_r$$

where $j^r \sim j_r$ *means that in the expansion of any symbolic power, any power* j^r *must be replaced by* j_r.

$$j_n = \tfrac{n(n-1)\ldots(n-k)}{k!}(j - j_0)^k \quad j^r \sim \tfrac{j_r}{n-r}$$

where $j^r \sim \frac{j_r}{n-r}$ *means that in the expansion of any symbolic power, any power* j^r *must be replaced by* $\frac{j_r}{n-r}$.

In the general case of a rational lattice polyhedron, the determination of the pseudo-polynomial of the enumerator needs the use of some more Ehrhart's results.

[1] The term *significant* means that P_1 defines a bounded polyhedral domain.

Theorem 3 *Let P_n be a rational k-polyhedron 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.

Computing the coefficients in the Ehrhart polynomial can be done through several approaches. For example, Barvinok[2] in [2] proposes a complexity study of the computation of these coefficients in terms of computation of the volumes of faces, since any coefficient can be related to volumes or surface areas of P_n. But since the problem of volume computation is NP-hard [4], results given by Barvinok step on the strong assumption of a *Volume Computation Oracle*.

We rather choose a more computable method. Due to this last theorem, the periodic coefficients are known. Since the period is equal to the denominator of the k-polyhedron P_n and the degree is equal to k, the determination of the Ehrhart pseudo-polynomial can be done by simply resolving a system of rational linear equations. Since the number of equations ne of the system must be equal to the number of unknown values, the complexity depends on the structure of P_n, its dimension, and on the size of its denominator. Moreover, ne initial values of the enumerator have to be counted.

All these results concern normal polyhedra defined by a combination of linear constraints. When considering some nonlinear ones, the transformation rules given in introduction have to be applied first.

3 Applications to the analysis of scientific programs

3.1 The general problem

The use of Ehrhart's results needs first the computation of the vertices of the considered normal polyhedron P_n. We use the algorithm of Loechner and Wilde [13] which determines the parametric coordinates of the vertices of any convex polytope and their associated definition domains. This algorithm is connected to a software package called *polyhedron* [23] which eliminates redundant constraints. Two particular problems have to be considered:

- P_n has to be homothetic (or bordered) according to definitions 2 and 3, i.e. P_n (or its kernel) can be deduced from P_1 (or the kernel of P_1) by the n-homothetic transformation. But in practice, P_1 does not correspond to the lowest possible size of P_n, since the initial value of n is depending on the constraints C_k and on some definition domains of these constraints. For example, let us consider the set of constraints defined by $\{4 \leq i \leq 2n, i \leq n+10, 1 \leq j \leq i\}$. The Loechner/Wilde algorithm computes the following answer:
  - if $n \geq 2$ and $n \leq 10$, the vertices are $(2n, 1)$, $(2n, 2n)$, $(4, 1)$, $(4, 4)$.
  - if $n > 10$, the vertices are $(n+10, 1)$, $(n+10, n+10)$, $(4, 1)$, $(4, 4)$.

  This result shows that the lowest value of n is 2 and hence, P_1 is undefined. This problem can be avoided by simply substituting n with $m = n - 1$ and considering $P_m = \{4 \leq i \leq 2m+2, i \leq m+11, 1 \leq j \leq i\}$. Moreover, this example shows that generally, several definition domains of n have to be considered.
- Due to the geometric modeling, as illustrated in the next section, 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 Loechner/Wilde 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 $P_n^j, j \neq i$.
  - 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.

3.2 Examples and related work

All the examples considered in this section come from several related works [11, 12, 22, 21, 9, 16] and were also handled by W. Pugh in [16], except example 7 whose aim is to show the generality of our method.

Example 1 M. Haghighat and C. Polychronopoulos present in [11, 12] a method for volume computation. Their first example is $\sum_{i=1}^{n} \sum_{j=3}^{i} \sum_{k=j}^{5} 1$. This sum defines a set of linear constraints $\{1 \leq i \leq n, 3 \leq j \leq i, j \leq k \leq 5\}$. The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope P_n:

- (1) if $3 \leq n \leq 5$, the vertices are $(3,3,5)$, $(3,3,3)$, $(n,3,5)$, $(n,3,3)$, $(n,n,5)$, (n,n,n).
- (2) if $n > 5$, the vertices are $(3,3,5)$, $(3,3,3)$, $(n,3,5)$, $(n,3,3)$, $(5,5,5)$, $(n,5,5)$.

Hence, P_n is an integral 3-polytope and the associated Ehrhart polynomial $j_n = c_1 n^2 + c_2 n + c_3$ is computed on each definition domain (1) and (2):

From $n = 3$ to 5 we count $j_n = 3, 8, 14$.

$$(1) \begin{cases} 9c_1 + 3c_2 + c_3 & = j_3 = 3 \\ 16c_1 + 4c_2 + c_3 & = j_4 = 8 \\ 25c_1 + 5c_2 + c_3 & = j_5 = 14 \end{cases}$$

This system results in $j_n = \frac{1}{2}n^2 + \frac{3}{2}n - 6$. This first polynomial is defined at three points instead of two as computed by Pugh in [16]. That is the reason why his symbolic result is different.

From $n = 6$ to 8 we count $j_n = 20, 26, 32$.

$$(2) \begin{cases} 36c_1 + 6c_2 + c_3 & = j_6 = 20 \\ 49c_1 + 7c_2 + c_3 & = j_7 = 26 \\ 64c_1 + 8c_2 + c_3 & = j_8 = 32 \end{cases}$$

This system results in $j_n = 6n - 16$, which is the same result as W. Pugh. M. Haghighat and C. Polychronopoulos derive an answer of

$$\mu(min(n-2,3))(-(min(n,5))^3 + 15(min(n,5))^2 \\ -38min(n,5) + 24)/6 + 6max(n-5,0)$$

where $\mu(x)$ is defined to be 1 if x is positive and 0 otherwise. The answer they derive gives the same values as ours, but the form is quite different because of the min and max expressions they introduce. The results tend to be much more complicated. Anyway, our method does not generate min's and max's. □

Example 2 The second example in [11, 12] is

$$\sum_{i=1}^{2n} \sum_{j=1}^{min(i,2n-i)} 1$$

[2] Barvinok also proposed in [1] a polynomial-time algorithm for counting integral points in non-parametric polyhedra, i.e. when the dimension is fixed.

This sum defines a set of linear constraints $\{1 \leq i \leq 2n, 1 \leq j \leq i, i+j \leq 2n\}$. The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope P_n: if $n \geq 1$, the vertices are $(n,n), (2n-1,1), (1,1)$. Hence, P_n is an integral 2-polytope and the associated Ehrhart polynomial $j_n = c_1 n^2 + c_2 n + c_3$ is computed:

From $n = 1$ to 3 we count $j_n = 1, 4, 9$.

$$\begin{cases} c_1 + c_2 + c_3 & = j_1 = 1 \\ 4c_1 + 2c_2 + c_3 & = j_2 = 4 \\ 9c_1 + 3c_2 + c_3 & = j_3 = 9 \end{cases}$$

This system results in $j_n = n^2$, which is the same result as W. Pugh. In [11, 12], Haghighat and Polychronopoulos do not describe their technique in detail. They define a number of rules in order to transform expressions, but nothing is said on how to decide which rule has to be applied and when. As in [22], they assume the summation must be performed in a predetermined order. □

Example 3 We now handle Pugh's "more elaborate" example of [16], and show that our method does not meet any particular difficulty, while Pugh introduces some additional techniques which are not elaborated in his paper. This example is the set of linear constraints $\{1 \leq i, j \leq n, 2i \leq 3j\}$.

The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope P_n: if $n \geq \frac{2}{3}$, the vertices are $(\frac{3n}{2}, n), (1, \frac{2}{3}), (1, n)$. Hence, P_n is a rational 2-polytope and the associated Ehrhart polynomial is a pseudo-polynomial of period 6. Since it is evident that the affine span of P_n contains at least one integral point ((1,1) for example), theorem 3 can be used in order to reduce the number of unknown coefficients: According to theorem 3, the coefficient of n^2 is constant, i.e. $j_n = c_1 n^2 + [c_2, c_3, c_4, c_5, c_6, c_7]n + [c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}]$.

From $n = 1$ to 13 we count j_n = 1, 4, 8, 14, 21, 30, 40, 52, 65, 80, 96, 114, 133.

$$\begin{cases} c_1 + c_2 + c_8 & = j_1 = 1 \\ 4c_1 + 2c_3 + c_9 & = j_2 = 4 \\ 9c_1 + 3c_4 + c_{10} & = j_3 = 8 \\ 16c_1 + 4c_5 + c_{11} & = j_4 = 14 \\ 25c_1 + 5c_6 + c_{12} & = j_5 = 21 \\ 36c_1 + 6c_7 + c_{13} & = j_6 = 30 \\ 49c_1 + 7c_2 + c_8 & = j_7 = 40 \\ 64c_1 + 8c_3 + c_9 & = j_8 = 52 \\ 81c_1 + 9c_4 + c_{10} & = j_9 = 65 \\ 100c_1 + 10c_5 + c_{11} & = j_{10} = 80 \\ 121c_1 + 11c_6 + c_{12} & = j_{11} = 96 \\ 144c_1 + 12c_7 + c_{13} & = j_{12} = 114 \\ 169c_1 + 13c_2 + c_8 & = j_{13} = 133 \end{cases}$$

This system results in

$$j_n = \frac{3}{4}n^2 + \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] n + \left[-\frac{1}{4}, 0, -\frac{1}{4}, 0, -\frac{1}{4}, 0\right]$$

which can be directly simplified to

$$j_n = \frac{3}{4}n^2 + \frac{1}{2}n + \left[-\frac{1}{4}, 0\right]$$

Pugh in [16] derives through his techniques an answer of

$$\begin{cases} \frac{(n+n \bmod 2)(3n+3(n \bmod 2)-2)}{8} & if\ 1 \leq n \\ \frac{3(n-n \bmod 2)(n-n \bmod 2+2)}{8} & if\ 2 \leq n \end{cases}$$

This expression is then intuitively simplified to $\frac{3n^2+2n-n \bmod 2}{4}$ giving the same answers as our Ehrhart polynomial. This example shows that Pugh's technique tends to complicated answers when considering rational polytopes, and hence, does not allow to consider systematically any set of rational linear constraints. Our method, without being too complicated, handles any set of rational linear constraints without any restriction, through underlying tools dealing with improved results on polyhedral combinatorics and linear programming.

The method described by Pugh in [16] consists in a set of techniques, each dedicated to simplify 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 needs the use of standard formulas for sums of powers (the author expects it will be sufficient to hard code the formulas for powers up to 10). All these facts clearly show some limits in the general use of these techniques. □

Example 4 Ferrante, Sarkar and Thrash in [9] present methods for computing the number of distinct memory locations and cache lines accessed by a loop nest. This information is useful in evaluating cache effectiveness. Their first example consists in calculating the number of distinct memory locations touched by

```
for i := 1 to 8 do
    for j := 1 to 5 do
        a(6i+9j-7) = a(6i+9j-7) + 5
```

This loop nest is not parametric and hence we prefer transforming it as a parametric loop nest

```
for i := 1 to n+3 do
    for j := 1 to n do
        a(6i+9j-7) = a(6i+9j-7) + 5
```

The number of distinct memory locations touched by the first form is then given with $n = 5$. This problem is equivalent to counting the number of distinct values of `6i+9j-7` s.t. $1 \leq i \leq n+3$ and $1 \leq j \leq n$. From a geometrical point of view, all the couples (i,j) resulting in the same value of $6i + 9j - 7$ all belong to a line generated by the vector $\binom{3}{-2}$. Hence, the initial problem is equivalent to counting the number of points resulting from the linear projection of all the (i,j)'s along this vector. The same type of resolution occurs when counting the number of needed processors from a linear mapping function, as shown in [3]. The used geometrical model is fully detailed in [3]. A short description is given in the following.

When projecting a convex polytope along a vector $\vec{p}$, the number of resulting points is equal to the number of extreme points on one of the extremities of each line segment generated by $\vec{p}$ on the polytope. These extreme points describe what we call *thick facets* of the polytope, which are also convex polytopes. The union of all these thick facets is not generally convex, but is a normal polyhedron, and our method can apply. Counting the number of points in this union of thick facets gives the number of points resulting from the projection along $\vec{p}$.

The vector $\vec{p} = \binom{3}{-2}$ defines two thick facets, which can be automatically determined [3], $P_n^1 = \{(i,j) \in \mathcal{Z}^2 | 1 \leq i \leq 3, 1 \leq j \leq n\}$ and $P_n^2 = \{(i,j) \in \mathcal{Z}^2 | 1 \leq i \leq n+3, n-1 \leq j \leq n\}$. The parametric vertices of $P_n^1 \cup P_n^2$ are computed by using the heuristic procedure described in subsection 3.1. It gives the following results: $(1,1)$, $(1,n)$, $(n+3,n)$, $(n+3,n-1)$, $(3,n-1)$ and $(3,1)$. Hence, $P_n^1 \cup P_n^2$ is an integral 2-polytope and the associated Ehrhart polynomial $j_n = c_1 n^2 + c_2 n + c_3$ is computed:

From $n = 3$ to 5 we count $j_n = 15, 20, 25$.

$$\begin{cases} 9c_1 + 3c_2 + c_3 & = j_3 = 15 \\ 16c_1 + 4c_2 + c_3 & = j_4 = 20 \\ 25c_1 + 5c_2 + c_3 & = j_5 = 25 \end{cases}$$

This system results in $j_n = 5n$ giving the same answer as Pugh for $n = 5$. □

Example 5 The second example in [9] is to calculate the number of memory locations touched in a Successive Over-Relaxation (SOR) code:

```
for i := 2 to n-1 do
  for j := 2 to n-1 do
      a(i,j) = (2*a(i,j) + a(i-1,j) + a(i+1,j)
                 + a(i,j-1) + a(i,j+1)) / 6
```

The elements of a touched by this loop are

$$\begin{array}{ll} & \{(i,j) | 2 \le i,j \le n-1\} \\ \cup & \{(i-1,j) | 2 \le i,j \le n-1\} \\ \cup & \{(i+1,j) | 2 \le i,j \le n-1\} \\ \cup & \{(i,j-1) | 2 \le i,j \le n-1\} \\ \cup & \{(i,j+1) | 2 \le i,j \le n-1\} \\ & \\ = & \{(i,j) | 2 \le i,j \le n-1\} \\ \cup & \{(i,j) | 1 \le i \le n-2, 2 \le j \le n-1\} \\ \cup & \{(i,j) | 3 \le i \le n, 2 \le j \le n-1\} \\ \cup & \{(i,j) | 2 \le i \le n-1, 1 \le j \le n-2\} \\ \cup & \{(i,j) | 2 \le i \le n-1, 3 \le j \le n\} \end{array}$$

The parametric vertices of this union of polytopes P_n are computed by using the heuristic procedure described in subsection 3.1. It gives the following results: $(1,2)$, $(2,1)$, $(1,n-1)$, $(2,n)$, $(n-1,n)$, $(n,n-1)$, $(n-1,1)$ and $(n,2)$. Hence, it is an integral 2-polytope and the associated Ehrhart polynomial $j_n = c_1 n^2 + c_2 n + c_3$ is computed:

From $n=4$ to 6 we count $j_n = 12, 21, 32$.

$$\begin{cases} 16c_1 + 4c_2 + c_3 & = j_4 = 12 \\ 25c_1 + 5c_2 + c_3 & = j_5 = 21 \\ 36c_1 + 6c_2 + c_3 & = j_6 = 32 \end{cases}$$

This system results in $j_n = n^2 - 4$ giving the same answer as Pugh.

To calculate the number of cache lines touched, we need a mapping from array elements to cache lines. We choose the same simple mapping as Pugh, which is to state that a reference to element $a(i,j)$ of an array references cache line $((i-1) \div 16, j)$. Hence, the problem consists in counting the number of distinct values of $((i-1) \div 16, j)$ for all the $a(i,j)$'s touched by the loop.

In such a case, the way of using our method is to model these different values by a union of polytopes U_n, deduced from the union of polytopes defining the elements touched by the loop P_n, and by computing the Ehrhart polynomial of U_n in a lattice $\mathcal{L}^2 \neq \mathcal{Z}^2$. This is done by transforming U_n through G^{-1}, where G is the dilatation matrix of $\mathcal{L}^2$, and then by computing the Ehrhart polynomial of $\left(G^{-1} \cdot U_n\right)$ in $\mathcal{Z}^2$.

On the affine span of the initial union of polytopes, the final result can be given by counting some points characterized by $(i-1)$ *mod* $16 = 0$, and such that $\exists \alpha \in [0..15]$, $\exists j$, $(i+\alpha, j) \in P_n$. All these points belong to the lattice defined by $\mathcal{L}^2 = \{\alpha_1 \binom{16}{0} + \alpha_2 \binom{0}{1} | \alpha_1, \alpha_2 \in \mathcal{Z}\}$.

Let $G = \begin{pmatrix} 16 & 0 \\ 0 & 1 \end{pmatrix}$. By moving P_n along the vector $\binom{-1}{0}$ and by applying G^{-1} to P_n, it is obtained a new union of polytopes containing only these interesting points (moved and transformed by G^{-1}), excepting those points of P_n characterized by $(i-1)$ *mod* $16 \neq 0$ and $\forall j$, $(i - ((i-1)\ mod\ 16), j) \notin P_n$. In order to get the whole union of polytopes U_n, points of the form $(i - ((i-1)\ mod\ 16), j)$ for all $(i,j) \in P_n$ have to be added to P_n. Hence, the definition of P_n is extended to U_n:

$$\begin{array}{ll} & \{(i,j) | 2 \le i,j \le n-1\} \\ \cup & \{(i,j) | 1 \le i \le n-2, 2 \le j \le n-1\} \\ \cup & \{(i,j) | 3 \le i \le n, 2 \le j \le n-1\} \\ \cup & \{(i,j) | 2 \le i \le n-1, 1 \le j \le n-2\} \\ \cup & \{(i,j) | 2 \le i \le n-1, 3 \le j \le n\} \\ \cup & \{(i - ((i-1)\ mod\ 16), j) | (i,j) \in P_n\} \end{array}$$

In order to compute the Ehrhart polynomial of $\left(G^{-1} \cdot U_n\right)$, just its vertices have to be determined. So there is no need to compute its constraints definition. It is sufficient to apply G^{-1} to the vertices of U_n to get the ones of $\left(G^{-1} \cdot U_n\right)$, since any polytope is entirely defined by its vertices.

$$\begin{cases} 16c_1 + 4c_5 + c_{21} & = j_4 = 4 \\ 25c_1 + 5c_6 + c_{22} & = j_5 = 5 \\ 36c_1 + 6c_7 + c_{23} & = j_6 = 6 \\ 49c_1 + 7c_8 + c_{24} & = j_7 = 7 \\ 64c_1 + 8c_9 + c_{25} & = j_8 = 8 \\ 81c_1 + 9c_{10} + c_{26} & = j_9 = 9 \\ 100c_1 + 10c_{11} + c_{27} & = j_{10} = 10 \\ 121c_1 + 11c_{12} + c_{28} & = j_{11} = 11 \\ 144c_1 + 12c_{13} + c_{29} & = j_{12} = 12 \\ 169c_1 + 13c_{14} + c_{30} & = j_{13} = 13 \\ 196c_1 + 14c_{15} + c_{31} & = j_{14} = 14 \\ 225c_1 + 15c_{16} + c_{32} & = j_{15} = 15 \\ 256c_1 + 16c_{17} + c_{33} & = j_{16} = 16 \\ 289c_1 + 17c_2 + c_{18} & = j_{17} = 32 \\ 324c_1 + 18c_3 + c_{19} & = j_{18} = 36 \\ 361c_1 + 19c_4 + c_{20} & = j_{19} = 38 \\ 400c_1 + 20c_5 + c_{21} & = j_{20} = 40 \\ 441c_1 + 21c_6 + c_{22} & = j_{21} = 42 \\ 484c_1 + 22c_7 + c_{23} & = j_{22} = 44 \\ 529c_1 + 23c_8 + c_{24} & = j_{23} = 46 \\ 576c_1 + 24c_9 + c_{25} & = j_{24} = 48 \\ 625c_1 + 25c_{10} + c_{26} & = j_{25} = 50 \\ 676c_1 + 26c_{11} + c_{27} & = j_{26} = 52 \\ 729c_1 + 27c_{12} + c_{28} & = j_{27} = 54 \\ 784c_1 + 28c_{13} + c_{29} & = j_{28} = 56 \\ 841c_1 + 29c_{14} + c_{30} & = j_{29} = 58 \\ 900c_1 + 30c_{15} + c_{31} & = j_{30} = 60 \\ 961c_1 + 31c_{16} + c_{32} & = j_{31} = 62 \\ 1024c_1 + 32c_{17} + c_{33} & = j_{32} = 64 \\ 1089c_1 + 33c_2 + c_{18} & = j_{33} = 97 \\ 1156c_1 + 34c_3 + c_{19} & = j_{34} = 102 \\ 1225c_1 + 35c_4 + c_{20} & = j_{35} = 105 \\ 1296c_1 + 36c_5 + c_{21} & = j_{36} = 108 \end{cases}$$

Figure 1: the linear equations system of example 5

The parametric vertices of U_n are computed: $(0,1)$, $(0,n)$, $(n-2,n)$, $(n-1,n-1)$, $(n-2,1)$ and $(n-1,2)$, which are transformed through G^{-1}, resulting in the parametric vertices of $\left(G^{-1} \cdot U_n\right)$: $(0,1)$, $(0,n)$, $(\frac{n-2}{16}, n)$, $(\frac{n-1}{16}, n-1)$, $(\frac{n-2}{16}, 1)$ and $(\frac{n-1}{16}, 2)$. Hence, it is a rational 2-polytope and the associated Ehrhart polynomial is a pseudo-polynomial of period 16. Theorem 3 is used to reduce the number of unknown coefficients:

$$\begin{array}{ll} j_n & = c_1 n^2 \\ & +[c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, \\ & c_{16}, c_{17}]n \\ & +[c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}, c_{27}, c_{28}, c_{29}, \\ & c_{30}, c_{31}, c_{32}, c_{33}] \end{array}$$

From $n=4$ to 36 we count $j_n = 4$, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 32, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 97, 102, 105, 108. The system, which is shown on figure 1, results in

$$\begin{array}{ll} j_n & = \frac{1}{16} n^2 \\ & +[\frac{15}{16}, \frac{7}{8}, \frac{13}{16}, \frac{3}{4}, \frac{11}{16}, \frac{5}{8}, \frac{9}{16}, \frac{1}{2}, \frac{7}{16}, \frac{3}{8}, \frac{5}{16}, \frac{1}{4}, \frac{3}{16}, \frac{1}{8}, \frac{1}{16}, 0]n \\ & +[-2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \end{array}$$

Pugh in [16] derives an answer, giving the same results as ours: $(\sum : n \geq 3 : n(1+(n-2) \div 16)) + (\sum : n\ mod\ 16 = 1 \wedge n \geq 17 : n-2)$, which means in a more common expression:

- if $n \geq 17$ and $n\ mod\ 16 = 0$, $j_n = n(1+(n-2) \div 16) + n - 2$
- if $n \geq 3$, $j_n = n(1+(n-2) \div 16)$

whose symbolic determination is not detailed in the paper. The method described in [9] by Ferrante et al., cannot handle unions of polytopes, does not compute symbolic answers, often computes only an approximation and uses expensive methods to handle the cache lines touched by a set of references. □

Example 6 An extension of our method to several size parameters is currently developped. It is applied to an example handled by Tawbi in [21] which is $\sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=j}^{m} 1$. This sum defines a set of linear constraints $\{1 \leq i \leq n, 1 \leq j \leq i, j \leq k \leq m\}$. The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope $P_{n,m}$:

- (1) if $1 \leq n \leq m$, the vertices are $(1,1,m)$, $(1,1,1)$, $(n,1,m)$, $(n,1,1)$, (n,n,m), (n,n,n).

- (2) if $1 \le m \le n$, the vertices are $(1,1,m)$, $(1,1,1)$, $(n,1,m)$, $(n,1,1)$, (m,m,m), (n,m,m).

Hence, $P_{n,m}$ is an integral 3-polytope and the associated Ehrhart polynomial is of the form $j_{n,m} = \sum_{i=0}^{3}\sum_{j=0}^{3} c_{ij} n^i m^j$ and has to be computed on each definition domain (1) and (2). Our method, applied with several size parameters, uses a symbolic resolution by first considering the polynomial $j_n = d_1 n^3 + d_2 n^2 + d_3 n + d_4$ where:

$$\begin{array}{rcl} d_1 &=& c_{33}m^3 + c_{32}m^2 + c_{31}m + c_{30} \\ d_2 &=& c_{23}m^3 + c_{22}m^2 + c_{21}m + c_{20} \\ d_3 &=& c_{13}m^3 + c_{12}m^2 + c_{11}m + c_{10} \\ d_4 &=& c_{03}m^3 + c_{02}m^2 + c_{01}m + c_{00} \end{array}$$

It follows the symbolic resolution:

$$\left\{ \begin{array}{ll} 64d_1 + 16d_2 + 4d_3 + d_4 & = j_4 = j_{4,m} \\ 125d_1 + 25d_2 + 5d_3 + d_4 & = j_5 = j_{5,m} \\ 216d_1 + 36d_2 + 6d_3 + d_4 & = j_6 = j_{6,m} \\ 343d_1 + 49d_2 + 7d_3 + d_4 & = j_7 = j_{7,m} \end{array} \right.$$

This system results in

$$\begin{array}{rl} j_n & = \frac{j_{7,m} - 3j_{6,m} + 3j_{5,m} - j_{4,m}}{6} n^3 \\ & + \frac{-5j_{7,m} + 16j_{6,m} - 17j_{5,m} + 6j_{4,m}}{2} n^2 \\ & + \frac{74j_{7,m} - 249j_{6,m} + 282j_{5,m} - 107j_{4,m}}{6} n \\ & -20j_{7,m} + 70j_{6,m} - 84j_{5,m} + 35j_{4,m} \end{array}$$

In order to evaluate these symbolic coefficients, $j_{4,m}$, $j_{5,m}$, $j_{6,m}$ and $j_{7,m}$ have to be computed on each definition domain (1) and (2):

For $n = 4$ and from $m = 4$ to 7, we count $j_{n,m} = 30, 40, 50, 60$.

$$\left\{ \begin{array}{ll} 64c_1 + 16c_2 + 4c_3 + c_4 & = j_{4,4} = 30 \\ 125c_1 + 25c_2 + 5c_3 + c_4 & = j_{4,5} = 40 \\ 216c_1 + 36c_2 + 6c_3 + c_4 & = j_{4,6} = 50 \\ 343c_1 + 49c_2 + 7c_3 + c_4 & = j_{4,7} = 60 \end{array} \right.$$

This system results in $j_{4,m} = 10m - 10$. In the same way, we compute $j_{5,m} = 15m - 20$, $j_{6,m} = 21m - 35$ and $j_{7,m} = 28m - 56$.

Hence, d_1, d_2, d_3 and d_4 can be computed as being: $d_1 = -\frac{1}{6}$, $d_2 = \frac{m}{2}$, $d_3 = \frac{3m+1}{6}$ and $d_4 = 0$. Finally, $j_{n,m}$ is evaluated for domain (1), i.e. $1 \le n \le m$, as being:

$$\begin{array}{rcl} j_{n,m} &=& -\frac{1}{6}n^3 + \frac{m}{2}n^2 + \frac{3m+1}{6}n + 0 \\ &=& -\frac{n^3}{6} + \frac{n^2 m}{2} + \frac{nm}{2} + \frac{n}{6} \end{array}$$

The same work is done for domain (2), i.e. $1 \le m \le n$, resulting in $j_{n,m} = -\frac{m^3}{6} + \frac{m^2 n}{2} + \frac{mn}{2} + \frac{m}{6}$, which is the same result as Pugh. Our method can be applied in the same way when considering several size parameters in the case of rational polytopes, case which is not systematically handled by Pugh's techniques (example 5). Tawbi in [22, 21] first uses a polyhedral splitting technique to characterize the definition domains, and an elimination of the variables in a predetermined order. However, her method does not allow to compute systematically exact answers as ours. □

Example 7 We propose a more elaborate example, requiring to utilize entirely the method we have described. This example involves rational parameterized polytopes depending on three size parameters. 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
      a(i,j,k) = a(i,j,k-1) + a(k,j,k)*a(i,k,k)
```

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

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

The Loechner/Wilde 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

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

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,

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

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, $j(n, m, p)$ is determined by resolving the system shown on figure 2. The following result is obtained, giving the number of flops executed by the loop nest when $n > p$:

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

$$
\left\{
\begin{array}{l}
729c_1 + 81c_3 + 9c_5 + c_7 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{17}{4}, \frac{9}{2}]_m p^2 + [\frac{167}{12}, \frac{44}{3}]_m p + [\frac{19}{2}, 10]_m \\
1000c_2 + 100c_4 + 10c_6 + c_8 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{19}{4}, 5]_m p^2 + [\frac{185}{12}, \frac{97}{6}]_m p + [\frac{21}{2}, 11]_m \\
1331c_1 + 121c_3 + 11c_5 + c_7 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{21}{4}, \frac{11}{2}]_m p^2 + [\frac{203}{12}, \frac{53}{3}]_m p + [\frac{23}{2}, 12]_m \\
1728c_2 + 144c_4 + 12c_6 + c_8 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{23}{4}, 6]_m p^2 + [\frac{221}{12}, \frac{115}{6}]_m p + [\frac{25}{2}, 13]_m \\
2197c_1 + 169c_3 + 13c_5 + c_7 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{25}{4}, \frac{13}{2}]_m p^2 + [\frac{239}{12}, \frac{62}{3}]_m p + [\frac{27}{2}, 14]_m \\
2744c_2 + 196c_4 + 14c_6 + c_8 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{27}{4}, 7]_m p^2 + [\frac{257}{12}, \frac{133}{6}]_m p + [\frac{29}{2}, 15]_m \\
3375c_1 + 225c_3 + 15c_5 + c_7 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{29}{4}, \frac{15}{2}]_m p^2 + [\frac{275}{12}, \frac{71}{3}]_m p + [\frac{31}{2}, 16]_m \\
4096c_2 + 256c_4 + 16c_6 + c_8 \\
= \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{31}{4}, 8]_m p^2 + [\frac{293}{12}, \frac{151}{6}]_m p + [\frac{33}{2}, 17]_m
\end{array}
\right.
$$

Figure 2: the linear equations system of example 7

4 Conclusion

We have presented a general method, based on the vertices of parameterized polytopes and Ehrhart polynomials, which allows to compute the number of integer solutions of selected free variables in a set of linear and/or nonlinear contraints. Our answers are given symbolically, in terms of size parameters, having many applications in the analysis and transformation of scientific programs. This method is quite simple and handles any set of rational linear constraints without any restriction, through underlying tools dealing with improved results on polyhedral combinatorics and linear programming. Moreover, the method gives systematically exact answers. It is entirely computable and will be soon implemented to be connected to the parametric vertices finding algorithm of Loechner and Wilde [13]. A first application of our method, to the computation of the potential parallelism of a given parallel loop, and the number of needed processors from a linear mapping function, was already presented in [3]. Compared with methods which ensure more limited capabilities [22, 11, 16], this paper allows some important observations:

- summations over several variables should not presume an order in which to perform the summation (as pointed out by Pugh in [16]),
- a general technique must consider rational polytopes too, since the nested loops model yields rational affine loop bounds, and since it is useful to handle systematically a wide variety of applications in the parallelization of scientific programs.
- in order to be simple, the answers gain to be expressed without any complex function as *min*'s, *max*'s, *mod*'s, floors or ceilings. In this way, periodic numbers introduced by Ehrhart [8] are a good answer, and can undoubtly be used in some other context.

References

[1] A. I. Barvinok. A polynomial-time algorithm for counting integral points in polyhedra when the dimension is fixed. In *Proc. of the 34th Symp. on the Foundations of Computer Science (FOCS'93)*, pages 566-572. IEEE Computer Society Press, New York, 1993.

[2] A. I. Barvinok. Computing the Ehrhart Polynomial of a Convex Lattice Polytope. *Discrete Comput. Geom.*, 12:35-48,1994.

[3] Ph. Clauss. The volume of a lattice polyhedron to enumerate processors and parallelism. Research Report ICPS 95-11, Submitted to publication, 1995. `http://icps.u-strasbg.fr/pub-95/pub-95-11.ps.gz`

[4] M. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17(5):967-974, 1988.

[5] E. Ehrhart. Sur les polyèdres rationnels homothétiques à n dimensions. *C.R. Acad. Sci. Paris*, 254:616-618, 1962.

[6] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire I. *J. Reine Angew. Math.*, 226:1-29, 1967.

[7] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire II. *J. Reine Angew. Math.*, 227:25-49, 1967.

[8] E. Ehrhart. *Polynômes arithmétiques et Méthode des Polyèdres en Combinatoire.* International Series of Numerical Mathematics, vol.35, Birkhäuser Verlag, Basel/Stuttgart, 1977.

[9] 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.

[10] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers.* North-Holland, Amsterdam, 1987.

[11] M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. *U. Banerjee et al., ed., Languages and Compilers for Parallel Computing.* Springer-Verlag, Aug. 1993. LNCS 768, *Proc. of the 6th annual worshop on Programming Languages and Compilers for Parallel Computing.*

[12] M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. Technical Report 1317, CSRD, Univ. of Illinois, Aug. 1993.

[13] V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. Research Report ICPS 95-16, Submitted to publication, 1995. `http://icps.u-strasbg.fr/pub-95/pub-95-16.ps.gz`

[14] I. G. Macdonald. The volume of a lattice polyhedron. *Proc. Camb. Phil. Soc.*, 59:719-726, 1963.

[15] I. G. Macdonald. Polynomials associated with finite cell-complexes. *J. London Math. Soc.*, 4(2):181-192, 1971.

[16] W. Pugh. Counting Solutions to Presburger Formulas: How and Why. *Proc. of the 1994 ACM SIGPLAN Conf. on Programming Language Design and Implementation*, 1994.

[17] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, New York, 1986.

[18] H. S. Stone and D. Thiebaut. Footprints in the cache. *Proc. ACM SIGMETRICS 1986*, pp. 4-8, May 1986.

[19] R. P. Stanley. *Combinatorics and Commutative Algebra.* Birkhäuser, Boston, 1983.

[20] R. P. Stanley. *Enumerative Combinatorics Volume I.* Wadsworth & Brooks/Cole, Monterey, California, 1986.

[21] N. Tawbi. Estimation of nested loop execution time by integer arithmetics in convex polyhedra. *Proc. of the 1994 Int. Parallel Processing Symp.*, Apr. 1994.

[22] N. Tawbi and P. Feautrier. Processor allocation and loop scheduling on multiprocessor computers. *Proc. of the 1992 Int. Conf. on Supercomputing*, pages 63-71, July 1992.

[23] D. K. Wilde. A library for doing polyhedral operations. *Master's thesis*, Oregon State Univ., Corvallis, Oregon, december 1993. Also published as IRISA technical report PI 785, Rennes, France, dec. 1993.