Integer Affine Transformations of Parametric $\mathbb{Z}$ -polytopes and Applications to Loop Nest Optimization

RACHID SEGHIR, University of Batna, Algeria

VINCENT LOECHNER, INRIA, University of Strasbourg, France

BENOÎT MEISTER, Reservoir Labs, New York

The polyhedral model is a well-known compiler optimization framework for the analysis and transformation of affine loop nests. We present a new method to solve a difficult geometric operation that is raised by this model: the integer affine transformation of parametric $\mathbb{Z}$ -polytopes. The result of such a transformation is given by a worst-case exponential union of $\mathbb{Z}$ -polytopes. We also propose a polynomial algorithm (for fixed dimension), to count points in arbitrary unions of a fixed number of parametric $\mathbb{Z}$ -polytopes. We implemented these algorithms and compared them to other existing algorithms, for a set of applications to loop nest analysis and optimization.

Categories and Subject Descriptors: D.3.4 [Programming Languages]: Processors—Compilers, Optimization; G.2.1 [Discrete Mathematics]: Combinatorics—Counting problems

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Counting solutions to Presburger formulas, Ehrhart polynomials, memory hierarchy optimization, parametric $\mathbb{Z}$ -polytopes transformation and enumeration, polyhedral model

ACM Reference Format:

Seghir, R., Loechner, V., and Meister, B. 2012. Integer affine transformations of parametric $\mathbb{Z}$ -polytopes and applications to loop nest optimization. *ACM Trans. Architect. Code Optim.* 9, 2, Article 8 (June 2012), 27 pages.

DOI = 10.1145/2207222.2207224 <http://doi.acm.org/10.1145/2207222.2207224>

1. INTRODUCTION

Many affine loop nests analyses and optimizations raise the problem of counting the number of images by an affine integer transformation of the lattice points contained in a parametric polytope: array linearization for hardware design, cache accesses computation and cache optimization, memory size computation, data distribution and locality optimization, etc. This counting problem was raised in this area a twenty years ago [Pugh 1991, 1994]. Since then, many solutions have been proposed, but none of them conclusively solves it. The problem of counting solutions to a parametric Presburger formula is equivalent [Gupta and Rajopadhye 2007] to counting points in the image by integer affine transformations of a union of parametric $\mathbb{Z}$ -polytopes. A parametric $\mathbb{Z}$ -polytope is the intersection between a parametric polytope (a bounded polyhedron, defined by a set of constraints depending linearly on formal parameters) and an integer lattice.

A preliminary version of this work was presented at the CASES'06 conference.

Authors' addresses: R. Seghir, Département d'informatique, Université de Batna, Algeria; V. Loechner, LSIIT (UMR 7005), ICPS Pôle API, Bd Sébastien Brant, 67412 Illkirch, France; email: loechner@unistra.fr; B. Meister, Reservoir Labs, New York, NY.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2012 ACM 1544-3566/2012/06-ART8 \$10.00

DOI 10.1145/2207222.2207224 <http://doi.acm.org/10.1145/2207222.2207224>

Early solutions compute an approximation or an upper bound of this number of points [Creusillet and Irigoin 1996; Ancourt et al. 1997; Zhao and Malik 2000; Ramanujam et al. 2001]. Some solved the subproblem of counting the integer points in a polytope (or equivalently in a single $\mathbb{Z}$ -polytope), depending on one [Barvinok 1994a; De Loera et al. 2004] or many [Clauss and Loechner 1998; Verdoolaege et al. 2004; 2007; Baldoni-Silva et al. 2006] parameters. Then, some exact methods to count integer affine images of $\mathbb{Z}$ -polytopes were proposed for the nonparametric case [Parker and Chatterjee 2004; Boigelot and Latour 2004; Zhu et al. 2006]. Some applications to loop nest analysis and optimization were developed without solving the general-case problem, but many of them used approximations [Zhao and Malik 2000; Ramanujam et al. 2001], or were restricted as they did not support formal parameters [Zhu et al. 2006]. Other applications need the exact solution [Loechner et al. 2002; Turjan et al. 2005].

Pugh [1994] first proposed an algorithm to solve the general-case problem, based on the Fourier-Motzkin variable elimination [Dantzig and Eaves 1973]. However, it seems that no implementation is available, probably due to the complexity of this method: the number of *splinters* which have to be built is an exponential function of the coefficients of the formula. Moreover, it is not clear whether it is able to eliminate more than one existential variable without scanning an exponentially large number of polytopes.

Verdoolaege et al. [2005] proposed applying simple rewriting rules (removing existential variables that are unique or redundant, decomposition in *independent splits*) to a disjoint union of parametric sets computed using the Omega library [Pugh 1991]. If these rewriting rules fail, they solve a parametric integer linear programming problem (using the PIP library [Feautrier 1988]), like Boulet and Redon [1998]. But the input of their algorithm, the disjoint union of sets computed from a Presburger formula, is worst-case exponential; and PIP is worst-case exponential. They also do not compute the actual integer affine image of the union of parametric $\mathbb{Z}$ -polytopes, but count the integer points of an equivalent set: if the application needs not only the number of points, but also the image itself, it cannot be used.

Another algorithm for computing the integer projection of a polytope was first proposed by Barvinok and Woods [2003], and then extended to parametric case by Verdoolaege and Woods [2005]. Even if this algorithm is theoretically polynomial-time (for fixed dimension), we believe that it still remains unusable in practice, because it uses algorithmic flatness theory and iterated Boolean combinations of rational generating functions. Indeed, this algorithm has frequently been referred to as “unimplementable” [Köppe et al. 2008]. A first-ever preliminary implementation of Barvinok-Woods’s algorithm has been reported by Köppe et al. [2008]. But this implementation seems to be rather slow even for simple examples. The implementation itself is considered a challenge.

More recently, Lasaruk and Sturm [2007] proposed a framework consisting in a generalization of Presburger arithmetic, where the coefficients are arbitrary polynomials in nonquantified variables. The elimination of the existential variables may result in a large number of atomic formulas function of the parameters (even for small examples). Furthermore, the atomic formulas may still contain bounded quantifiers. In order to expand this result into disjunctions, the parameters have to be instantiated. Hence the final result could not be given in a symbolic form (as a function of the parameters).

Table I summarizes the well-known algorithms for solving integer point counting problems, classified according to their precision and the type of the problems they solve.

In this article, we propose two new algorithms: (i) a polynomial-time algorithm to count points in the union of a fixed number of parametric $\mathbb{Z}$ -polytopes of fixed dimension, (ii) an algorithm to compute the integer affine image of a union of parametric

Table I. Well-Known Algorithms Dealing with Integer Points Counting Problems

Counting algorithm	Precision	Parametric	Counting problem
[Creusillet and Irigoien 1996], [Ancourt et al. 1997], [Zhao and Malik 2000], [Ramanujam et al. 2001]	approximate	yes	image of a union of $\mathbb{Z}$ -polytopes
[Barvinok 1994a], [De Loera et al. 2004]	exact	one parameter	one $\mathbb{Z}$ -polytope
[Clauss and Loechner 1998], [Verdoolaege et al. 2004], [Baldoni-Silva et al. 2006]	exact	yes	one $\mathbb{Z}$ -polytope
[Barvinok and Woods 2003]	exact	no	image of one $\mathbb{Z}$ -polytope
[Parker and Chatterjee 2004], [Boigelot and Latour 2004], [Zhu et al. 2006]	exact	no	image of a union of $\mathbb{Z}$ -polytopes
[Gupta and Rajopadhye 2007]	exact	yes	computing the image of a union of $\mathbb{Z}$ -polytopes
[Pugh 1994], [Verdoolaege et al. 2005], [Lasaruk and Sturm 2007], [Köppe et al. 2008]	exact	yes	image of a union of $\mathbb{Z}$ -polytopes

$\mathbb{Z}$ -polytopes, in the form of a worst-case exponential union of parametric $\mathbb{Z}$ -polytopes. We also compare our implementation with other work that compute the exact result, are parametric (for many parameters) and general (integer transformation of unions of $\mathbb{Z}$ -polytopes), and show that it is efficient and can be used to optimize real applications.

We will provide the reader with some references on using our work in compiler design and program optimization. The class of *parametric affine loop nests* (programs based on affine references to arrays in affine bounded loop nests, containing unknown variables at compile time) has been tackled in the past by many researchers, since it occurs frequently and is resource-consuming, in applications like calculation-intensive scientific applications, digital audio/video, graphics, compression/decompression, radar and DSP, etc. In this context, array linearization for hardware design [Turjan et al. 2005], cache access optimization [Ghosh et al. 1999; D’Alberto et al. 2001; Loechner et al. 2002], memory size computation [Zhao and Malik 2000; Zhu et al. 2006], and data distribution for NUMA-machines [Heine and Slowik 2000], reduce to the problem of computing integer affine images of a union of parametric $\mathbb{Z}$ -polytopes and enumerating them.

In Seghir and Loechner [2006], we showed, through a detailed example, an application in the field of embedded systems using our algorithms. Namely, *array linearization* deals with two major issues of memory/cache footprint optimization: *memory compression* and *spatial locality enhancement*. The basic idea is the transformation of a multi-dimensional array into a one-dimensional array in which elements are stored in the same order as they are first accessed. Thereby, only array elements that are actually used are stored in the new array, leading to memory compression. Moreover, the sizes of access strides are reduced, which leads to spatial locality enhancement.

Other applications have been reported in economics and in mathematics, in the context of combinatorics, representation theory, statistics and discrete optimization (see De Loera et al. [2004] for references).

The article is organized as follows: after presenting some basic concepts in Section 2, we describe our algorithm for computing the integer affine image of a $\mathbb{Z}$ -polytope in Section 3. Section 4 presents our algorithm for counting points in arbitrary unions of parametric $\mathbb{Z}$ -polytopes. In Section 5 we compare our approach to related work and provide some experiences and applications to loop nest optimization. Finally, the conclusions are given in Section 6.

2. BASIC CONCEPTS

In this section we recall some polyhedral and lattice definitions as well as some theoretical results on counting lattice points in parametric polytopes.

Definition 2.1. A d -dimensional rational polyhedron $P \subset \mathbb{Q}^d$ is a set of rational d -dimensional vectors $\mathbf{x}$ defined by linear inequalities

$$P = \{\mathbf{x} \in \mathbb{Q}^d \mid A\mathbf{x} \geq \mathbf{c}\}, \quad (1)$$

with $A \in \mathbb{Z}^{m \times d}$ and $\mathbf{c} \in \mathbb{Z}^m$. The system $A\mathbf{x} \geq \mathbf{c}$ may contain pairs of inequalities that are equivalent to an equality (called an implicit equality, that can be computed by Polylib [Loechner 1999] for example). In such a case, the polyhedron is said *nonfull-dimensional*.

A bounded rational polyhedron is commonly called a *rational polytope*.

Definition 2.2. The *affine hull* of a set $S \subset \mathbb{Q}^d$ is the affine subspace $\{\lambda_1 x_1 + \dots + \lambda_k x_k \mid \{x_1, \dots, x_k\} \subset S, \lambda_i \in \mathbb{Q}, \sum_i \lambda_i = 1\}$. The dimension of an affine hull is the dimension of its vector subspace.

Definition 2.3. The *geometric dimension* of a rational polyhedron $P \subset \mathbb{Q}^d$ is the dimension of its affine hull. Equivalently, it is equal to d minus the number of linearly independent (implicit) equalities in the system $A\mathbf{x} \geq \mathbf{c}$.

Definition 2.4. A *rational parametric polytope* $P_{\mathbf{p}}$ with n parameters $\mathbf{p}$ is a set of rational d -dimensional vectors $\mathbf{x}$ defined by linear inequalities on $\mathbf{x}$ and $\mathbf{p}$

$$P_{\mathbf{p}} = \{\mathbf{x} \in \mathbb{Q}^d \mid A\mathbf{x} \geq B\mathbf{p} + \mathbf{c}\}, \quad (2)$$

with $A \in \mathbb{Z}^{m \times d}$, $B \in \mathbb{Z}^{m \times n}$ and $\mathbf{c} \in \mathbb{Z}^m$, and such that for each fixed value $\mathbf{p}_0$ of $\mathbf{p}$, $P_{\mathbf{p}_0}$ defines a (possibly empty) rational polytope in $\mathbb{Q}^d$.

Definition 2.5. An d -dimensional integer lattice is a subset of $\mathbb{Z}^d$ defined by integer linear combinations of linearly independent integer vectors, called lattice-generating vectors (or lattice basis), plus an affine part.

$$L = \{A\mathbf{x} + \mathbf{c} \mid \mathbf{x} \in \mathbb{Z}^d\}, \quad (3)$$

where A is an integer matrix whose column vectors are the generators of the lattice and $\mathbf{c}$ is a constant integer vector. We also denote such a lattice by $\mathcal{L}(A, \mathbf{c})$.

Definition 2.6. A $\mathbb{Z}$ -polytope is the intersection between a rational polytope and an integer lattice defined in the same space. It can also be seen as a regular subset of the integer points contained in a rational polytope.

Definition 2.7. A *standard $\mathbb{Z}$ -polytope* is the intersection between a rational d -polytope and the *standard* lattice $\mathbb{Z}^d$.

Definition 2.8. An *integer affine transformation* of a d -dimensional $\mathbb{Z}$ -polytope $\mathcal{Z} = P \cap L$ is the transformation of its points by an integer affine function:

$$\begin{aligned} T : \mathbb{Z}^d &\rightarrow \mathbb{Z}^k \\ \mathbf{x} &\mapsto A\mathbf{x} + \mathbf{c} \end{aligned} \quad (4)$$

where A is an integer matrix and $\mathbf{c}$ an integer vector.

Definition 2.9. A *Presburger formula* is a set of linear equalities and inequalities on some integer variables, linked by the logical operators ($\neg$, $\wedge$, $\vee$), and the universal and existential quantifiers ($\forall$, $\exists$).

Definition 2.10. A *parametric Presburger formula* is a Presburger formula in which the constraints linearly depend on a vector $\mathbf{p}$ of symbolic integer parameters.

2.1. Ehrhart Theory

Definition 2.11. A rational n -periodic function¹ $U(\mathbf{p})$ is a function $\mathbb{Z}^n \rightarrow \mathbb{Q}$, such that there exists a *period* $\mathbf{q} = (q_1, \dots, q_n) \in \mathbb{N}^{+n}$, with $U(\mathbf{p}) = U(\mathbf{p}')$ whenever $p_i \equiv p'_i \pmod{q_i}$, for $1 \leq i \leq n$.

A periodic function can be represented by an n -dimensional lookup-table $U_{\mathbf{p}}$ such that $U(\mathbf{p}) = U_{\mathbf{p}}[p_1 \bmod q_1, \dots, p_n \bmod q_n]$. Such a lookup-table is typically written as a list or matrix of values enclosed in square brackets. For instance, $U_N = [1, \frac{3}{4}]_N$ is a 1-periodic function with period $q_1 = 2$; $U_N = 1$ if $N \bmod 2 \equiv 0$ and $U_N = \frac{3}{4}$ if $N \bmod 2 \equiv 1$. A 2-periodic function of period $(2, 3)$ can be represented as $\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{N,M} = [[a, b, c]_M, [d, e, f]_M]_N$, which is equal to b if $N = 0$ and $M = 1$, and so on.

The lookup-table representation of periodic functions has the advantage that it can be fully simplified, but it is worst case exponential [Verdoolaege et al. 2004, 2007]. Other representations avoid representing periodic functions with an exponential number of coefficients exist, such as integer parts, modulo and fractional representations. For instance, the lookup-table periodic function $U_N = [1, \frac{3}{4}]_N$ is equivalent to the following functions: $1 - \frac{N}{4} - \frac{1}{2} \lceil -\frac{N}{2} \rceil$, $1 - \frac{N}{4} + \frac{1}{2} \lfloor \frac{N}{2} \rfloor$, $1 - \frac{1}{4}(N \bmod 2)$ or $1 - \frac{1}{2} \{ \frac{N}{2} \}$, where $\{ \frac{N}{2} \}$ is the fractional part of $\frac{N}{2}$.

Definition 2.12. A *quasi-polynomial* of degree d in n variables $\mathbf{p} = (p_1, p_2, \dots, p_n)$ is a polynomial expression of degree d in $\mathbf{p}$ over the rational n -periodic functions

$$\sum_{\mathbf{i} \in I} U_{\mathbf{i}}(\mathbf{p}) \mathbf{p}^{\mathbf{i}}, \quad (5)$$

with $I \subset \mathbb{N}^n$, $\mathbf{i} = (i_1, \dots, i_n)$, $\sum_{k=1}^n i_k \leq d$, $U_{\mathbf{i}}$ periodic functions and $\mathbf{p}^{\mathbf{i}} = p_1^{i_1} p_2^{i_2} \dots p_n^{i_n}$. The *period* of a quasi-polynomial is the componentwise least common multiple (lcm) of the periods of its coefficients $U_{\mathbf{i}}$.

Definition 2.13. The *enumerator* $\mathcal{E}(P_{\mathbf{p}})$ of a parametric polytope $P_{\mathbf{p}}$ (or a parametric standard $\mathbb{Z}$ -polytope $\mathcal{Z} = P_{\mathbf{p}} \cap \mathbb{Z}^d$) is a function from the set of n -dimensional integer vectors $\mathbb{Z}^n$ to the set of natural numbers $\mathbb{N}$. The function value at $\mathbf{p}_0$, denoted $\mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0)$, is the number of integer points in polytope $P_{\mathbf{p}}$:

$$\begin{aligned} \mathcal{E}(P_{\mathbf{p}}) : \mathbb{Z}^n &\rightarrow \mathbb{N} \\ \mathbf{p}_0 &\mapsto \mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0) \end{aligned}$$

$$\mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0) = |\{\mathbf{x} \in \mathbb{Z}^d \mid A\mathbf{x} \geq B\mathbf{p}_0 + \mathbf{c}\}|,$$

where $|\cdot|$ denotes the cardinality.

We first consider a special case of parametric polytopes $P_{\mathbf{p}}$ that can be written as the convex combination of a fixed set of parametric vertices, where each vertex is an affine function of the parameters:

$$P_{\mathbf{p}} = \left\{ \mathbf{x} \in \mathbb{Q}^d \mid \mathbf{x} = V(\mathbf{p})\lambda, 0 \leq \lambda_j, \sum_j \lambda_j = 1 \right\}, \quad (6)$$

¹Also called periodic number in other work.

where the columns $V_j(\mathbf{p})$ of $V(\mathbf{p})$ are the vertices of $P_{\mathbf{p}}$, that is, $V_j(\mathbf{p}) = G_j\mathbf{p} + \mathbf{h}_j$ for some $G_j \in \mathbb{Q}^{d \times n}$ and $\mathbf{h}_j \in \mathbb{Q}^d$. Ehrhart [1977] showed that the enumerator of such a set can be represented as a *quasi-polynomial* function of $\mathbf{p}$, as defined by (5).

Loechnner, Wilde, and Clauss showed [Loechnner and Wilde 1997; Clauss and Loechnner 1998] that the parameter space of a general parametric polytope $P_{\mathbf{p}}$ as defined by (2) can be divided into a set of chambers (also called “validity domains”: a union of adjacent polytopes covering the parameter space), such that in each chamber, $P_{\mathbf{p}}$ has a fixed set of parametric vertices that are affine combinations of the parameters. In each chamber, the parametric polytope can therefore be written as (6). Using this decomposition, they proved the following theorem [Clauss and Loechnner 1998].

THEOREM 2.14. *The enumerator of a parametric polytope $P_{\mathbf{p}}$ of geometric dimension d is a quasi-polynomial (commonly known as Ehrhart quasi-polynomial) of degree d in $\mathbf{p}$ on each chamber. The componentwise period of the quasi-polynomial in a given chamber divides the componentwise lcm of the denominators that appear in the vertices defined on that chamber.*

We recently proposed a polynomial algorithm (for fixed dimension) [Verdoolaege et al. 2004, 2007] that counts lattice points in general parametric polytopes. This algorithm combines Barvinok’s counting method [Barvinok 1994b] with Clauss and Loechnner’s chamber decomposition [Clauss and Loechnner 1998]. We also use this algorithm to count points in integer affine transformations of parametric $\mathbb{Z}$ -polytopes.

3. INTEGER AFFINE TRANSFORMATIONS OF PARAMETRIC $\mathbb{Z}$ -POLYTOPES

This section presents our algorithm for computing the integer affine transformation of a parametric $\mathbb{Z}$ -polytope as a union of $\mathbb{Z}$ -polytopes. This is also called a linearly bounded lattice, or LBL [Teich and Thiele 1993]. The calculation of the number of points in such a union is described in Section 4.

It has been shown that the integer affine transformation of a $\mathbb{Z}$ -polytope can be written as a Presburger formula [Pugh 1994; Kreisel and Krevine 1967; Gupta and Rajopadhye 2007]:

Let $\mathcal{Z} = P_{\mathbf{p}} \cap L$ be a parametric $\mathbb{Z}$ -polytope, $P_{\mathbf{p}} = \{\mathbf{x} \in \mathbb{Q}^d \mid \mathbf{A}\mathbf{x} \geq \mathbf{B}\mathbf{p} + \mathbf{c}\}$ a parametric rational polytope, $L = \{\mathbf{A}'\mathbf{z} + \mathbf{c}' \mid \mathbf{z} \in \mathbb{Z}^d\}$ a lattice and

$$\begin{aligned} T : \mathbb{Z}^d &\rightarrow \mathbb{Z}^k \\ \mathbf{x} &\mapsto \mathbf{x}' = \mathbf{A}''\mathbf{x} + \mathbf{c}'' \end{aligned}$$

be an integer transformation. Then the transformation of $\mathcal{Z}$ by T is:

$$T(\mathcal{Z}) = \{\mathbf{x}' \in \mathbb{Z}^k \mid \exists \mathbf{x}, \mathbf{z} \in \mathbb{Z}^d, \mathbf{A}\mathbf{x} \geq \mathbf{B}\mathbf{p} + \mathbf{c}, \mathbf{x} = \mathbf{A}'\mathbf{z} + \mathbf{c}', \mathbf{x}' = \mathbf{A}''\mathbf{x} + \mathbf{c}''\} \quad (7)$$

which is a parametric Presburger formula. The other side of equivalence (from a Presburger formula to the transformation of a union of $\mathbb{Z}$ -polytopes) has been proved by Gupta and Rajopadhye [2007].

Example 3.1. The transformation of the parametric ($N \in \mathbb{Z}$ is a parameter) standard $\mathbb{Z}$ -polytope

$$\mathcal{Z}_N = \{(i, j, k) \in \mathbb{Z}^3 \mid 1 \leq i \leq N \wedge 1 \leq j \leq N \wedge 1 \leq k \leq N\}$$

by the integer affine function $T(i, j, k) = (3i + 6k, 5i + 2j + 1)$ can be written by adding two variables $x = 3i + 6k$ and $y = 5i + 2j + 1$, as the Presburger formula:

$$\begin{aligned} S = \{(x, y) \in \mathbb{Z}^2 \mid \exists (i, j, k) \in \mathbb{Z}^3 : \\ 1 \leq i \leq N \wedge 1 \leq j \leq N \wedge 1 \leq k \leq N \wedge x = 3i + 6k \wedge y = 5i + 2j + 1\}. \end{aligned}$$

The problem of calculating the integer affine transformation of a $\mathbb{Z}$ -polytope reduces to the elimination of the existential variables from a Presburger formula (Equation (7)). In such a formula, the conjunction of the equalities and inequalities define a non-full-dimensional standard $\mathbb{Z}$ -polytope $\hat{Z}$ in the combined space $\mathbb{Z}^{d'+d}$ (the cartesian product of the $\mathbb{Z}$ -polytope space $\mathbb{Z}^d$ and the existential variables space $\mathbb{Z}^{d'}$). A common way to eliminate the existential variables of the Presburger formula is to project them out of $\hat{Z}$. Our algorithm performs this in two steps. It starts by eliminating a first set of existential variables using equalities, as described in the next section. Then, it eliminates the remaining existential variables using inequalities of $\hat{Z}$.

3.1. Non-Full-Dimensional $\mathbb{Z}$ -Polytope Preprocessing

When removing existential variables using a set of equalities in $\mathbb{Z}$ -polytope $\hat{Z}$, one must ensure that there exist integer values of the variables to be eliminated for each integer value of the other variables. Without loss of generality, we consider the parameters as regular variables. Indeed, removing equalities from a *parametric* $\mathbb{Z}$ -polytope is done in the same way as for a nonparametric $\mathbb{Z}$ -polytope: parameters are free variables, and they are never eliminated whatever the number of equalities. We also assume that the $\mathbb{Z}$ -polytope is *standard*, since any arbitrary $\mathbb{Z}$ -polytope $Z = P \cap L$ can be transformed into a standard $\mathbb{Z}$ -polytope containing the same number of integer points. This is done by computing the preimage of polytope P by a matrix defining lattice L . Notice that it is possible to rewrite the final result as a function of the original variables by doing the inverse transformation, as explained in Section 4.

Consider a $(d' + d)$ -dimensional standard $\mathbb{Z}$ -polytope $\hat{Z}$, defined by a nonredundant system of linear constraints, with d' existential variables and d free variables. Let m be the number of equalities implying existential variables. The system of equalities is upper-triangularized, with the existential variables on the first columns. Let us denote the set of equalities implying existential variables by $E(\mathbf{y}', \mathbf{y})$, and the set of remaining constraints by $\bar{E}(\mathbf{y}', \mathbf{y})$, where $\mathbf{y}'$ is a vector of m variables chosen among the d' existential variables, and $\mathbf{y}$ is a vector of the k remaining variables, with $k = d' + d - m$.

In order to eliminate the m equalities, it suffices to solve the system of Diophantine equations $E(\mathbf{y}', \mathbf{y})$ in $\mathbf{y}'$ as a function of $\mathbf{y}$, and to substitute the result in the system of remaining constraints $\bar{E}(\mathbf{y}', \mathbf{y})$. The resulting polytope Y has then to be intersected with an integer lattice defining the valid values of $\mathbf{y}$. We call this lattice the *validity lattice* of Y . The result is $\mathbb{Z}$ -polytope $Z' = Y \cap L$. In the following, we explain how this validity lattice L is calculated.

The system of equalities $E(\mathbf{y}', \mathbf{y})$ can be rewritten as:

$$A\mathbf{y}' + B\mathbf{y} + \mathbf{c} = 0, \quad (8)$$

where A is a full row-rank $(m \times m)$ integer matrix, B is an $(m \times k)$ integer matrix and $\mathbf{c}$ is an integer² m -vector. We want to derive the valid values of $\mathbf{y}$ from Equation (8). In other words, we want a necessary and sufficient condition on $\mathbf{y}$ for an integer solution $\mathbf{y}'$ to exist.

Property 3.2. The integer values of $\mathbf{y}$ for which the system of equalities (8) admits an integer solution in $\mathbf{y}'$, if it exists, are given by a k -dimensional integer lattice $\mathcal{L}(G, \mathbf{y}_0)$. We call $\mathcal{L}(G, \mathbf{y}_0)$ the validity lattice over $\mathbf{y}$ induced by (8).

This property is based on the lattice intersection computation [Quinton et al. 1997]. Indeed, the necessary and sufficient condition on $\mathbf{y}$ for the existence of an integer

²Notice that these matrices and vector can originally be rational, in which case the lines are multiplied by the *lcm* of their denominators.

solution in $\mathbf{y}'$ to (8) is the following:

$$\exists \mathbf{z} \in \mathbb{Z}^m, \mathbf{A}\mathbf{y}' = -\mathbf{B}\mathbf{y} - \mathbf{c} = \mathbf{z},$$

or equivalently that there is an integer solution in $(\mathbf{y}', \mathbf{y}, \mathbf{z})$ to the system of equalities:

$$\begin{pmatrix} -A & 0 & I_m \\ 0 & B & I_m \end{pmatrix} \begin{pmatrix} \mathbf{y}' \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{c} \end{pmatrix}, \quad (9)$$

where I_m is the $(m \times m)$ identity matrix.

The left hand-side matrix in Equation (9) is, by construction, a $(2m \times (2m + k))$ full row-rank integer matrix. By Corollary 5.3c of Schrijver [1986] (p. 58) a feasible system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ has an integer solution as: $\mathbf{x} = \mathbf{S}\mathbf{t} + \mathbf{s}_0$. Hence, the integer solution to (9), if it exists, is given as a function of k free variables:

$$\begin{pmatrix} \mathbf{y}' \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} E \\ G \\ F \end{pmatrix} \mathbf{t} + \begin{pmatrix} \mathbf{y}'_0 \\ \mathbf{y}_0 \\ \mathbf{z}_0 \end{pmatrix}, \quad \mathbf{t} \in \mathbb{Z}^k. \quad (10)$$

The condition on $\mathbf{y}$ for the existence of an integer solution to (10) is then:

$$\mathbf{y} = \mathbf{G}\mathbf{t} + \mathbf{y}_0, \quad \mathbf{t} \in \mathbb{Z}^k,$$

where G is a $(k \times k)$ integer matrix and $\mathbf{y}_0$ is an integer k -vector. Hence, $\mathbf{y}$ should be part of the k -dimensional lattice $\mathcal{L}(G, \mathbf{y}_0)$.

Example 3.3. Consider the Presburger formula:

$$S = \{(x, y, z) \in \mathbb{Z}^3 \mid \exists (i, j) \in \mathbb{Z}^2 : 1 \leq i \leq N \wedge 1 \leq j \leq N \wedge 1 \leq x \leq N \wedge 3i + 6j = y \wedge 5i = -2x + z - 1\}. \quad (11)$$

In order to eliminate the equalities of this formula it suffices to eliminate variables i and j (as in the rational case), and to intersect the result with the validity lattice corresponding to values of x, y, z , and N for which the system of equalities admits an integer solution for (i, j) . The rational elimination of the equalities results in a full-dimensional polytope:

$$Y = \{(x, y, z) \in \mathbb{Q}^3 \mid 27 \leq 6x + 5y - 3z \leq 30N - 3 \wedge 6 \leq -2x + z \leq 5N + 1 \wedge 1 \leq x \leq N\}.$$

However, not all integer values of (x, y, z) in this set correspond to integer values of (i, j) . The validity lattice of this polytope is obtained by solving the system of equalities:

$$\left(\begin{array}{cc|cccc|cc} -3 & -6 & 0 & 0 & 0 & 0 & 1 & 0 \\ -5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & -1 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} i \\ j \\ x \\ y \\ z \\ N \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

The solution of this system can be computed as:

$$\begin{pmatrix} i \\ j \\ x \\ y \\ z \\ N \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 5 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Hence,

$$\begin{pmatrix} x \\ y \\ z \\ N \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

which defines an integer lattice $\mathcal{L}(G, \mathbf{y}_0) = \mathcal{L} \left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right)$.

The final result of eliminating equalities of formula (11) is the $\mathbb{Z}$ -polytope $\mathcal{Z}' = Y \cap \mathcal{L}(G, \mathbf{y}_0)$.

To continue the computation, let us call $\mathcal{Z}' = Y \cap L$ the resulting $\mathbb{Z}$ -polytope after eliminating the equalities implying existential variables in a $\mathbb{Z}$ -polytope $\hat{\mathcal{Z}}$. Two cases can occur.

- There are no remaining existential variables (it occurs when the number of existential variables is smaller or equal to the number of equalities that involve them, as in the example presented above). Then, $\mathcal{Z}'$ is the solution.
- There still remain existential variables to be eliminated in $\mathcal{Z}'$. In this case, we first reorganize the lattice so that its basis matrix becomes upper triangular. This is performed by computing its Hermite normal form. We then calculate the compression of $\mathbb{Z}$ -polytope $\mathcal{Z}'$, that is, the preimage of Y by the matrix defining lattice L . Finally, we eliminate the remaining existential variables as explained in Section 3.3. In order to preserve the original coordinates, we calculate the transformation of the resulting $\mathbb{Z}$ -polytopes by the submatrix obtained by removing the lines and columns corresponding to existential variables from the matrix defining L . The result is then intersected with the sublattice defined by this latter submatrix to obtain the desired transformation.

In the following, we consider without loss of generality, that our formulas do not contain equalities implying existential variables, and we focus on eliminating the remaining existential variables.

3.2. Existential Variable Elimination and Integer Projection

A classical algorithm to eliminate variables from a system of constraints is the Fourier-Motzkin elimination procedure [Dantzig and Eaves 1973]. It allows the elimination of a *rational* existential variable from a system of affine inequalities, defined on the set of rational numbers. Its main idea consists in rewriting the original system as a set of lower and upper bounds on the variable to be eliminated. Then, each pair of lower and upper bounds of the form $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$ is to be replaced by $\alpha l(\mathbf{x}, \mathbf{p}) \leq \beta u(\mathbf{x}, \mathbf{p})$, where z is the existential variable chosen to be eliminated, $l(\mathbf{x}, \mathbf{p})$

Fig. 1. The integer projection of a $\mathbb{Z}$ -polytope.

and $u(\mathbf{x}, \mathbf{p})$ are affine functions of variables $\mathbf{x}$ and parameters $\mathbf{p}$ independent of z , and α and β are strictly positive integer constants. This procedure has been extended to *integer* existential variable elimination by Pugh [1991, 1994] and Pugh and Wonnacott [1994] as follows:

Any pair of lower and upper bounds $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$ defines:

- an *exact shadow*, corresponding to the rational projection of the points belonging to this pair of constraints. This is given by $\alpha l(\mathbf{x}, \mathbf{p}) \leq \beta u(\mathbf{x}, \mathbf{p})$.
- a *dark shadow*, corresponding to the convex part of the exact shadow in which any integer point has at least one integer preimage. This is given by $\alpha l(\mathbf{x}, \mathbf{p}) + (\alpha - 1)(\beta - 1) \leq \beta u(\mathbf{x}, \mathbf{p})$. Notice that if $\alpha = 1$ or $\beta = 1$, the dark shadow is equal to the exact shadow.

The part of the exact shadow that does not belong to the dark shadow may contain integer points having integer preimages, and other integer points called *holes* having only rational preimages in $P_{\mathbf{p}}$ (see Figure 1).

The Omega test [Pugh 1991] answers the question: “is there an integer point in the projection having at least an integer preimage?” as follows.

- If the exact shadow does not contain any integer point, the answer is no.
- If the dark shadow contains at least one integer point, the answer is yes.
- Otherwise, the answer is not obvious. In this case, we have to know whether the part of the exact shadow which does not belong to the dark shadow contains an integer point having integer preimage(s) in $P_{\mathbf{p}}$.

In order to answer this latter question, Pugh and Wonnacott [1994] check whether the intersection between a certain number (function of α and β) of hyperplanes and the constraints of the original system contains an integer point (see Section 5).

Note 3.4. The integer projection of a $\mathbb{Z}$ -polytope results in a unique dark shadow (whatever the number of existential variables) and a union of $\mathbb{Z}$ -polytopes. Hence, the challenge is that of computing such a union.

3.3. Our Projection Method

In this section, we present the heart of our algorithm: the projection of a single pair of lower and upper bounds of the existential variable chosen to be eliminated. Indeed, the projection of the whole $\mathbb{Z}$ -polytope is obtained by intersecting the projections of all its pairs of bounds (like the well-known Fourier-Motzkin algorithm). The result is also intersected with the constraints that are independent of the eliminated variable.

Consider a pair of lower and upper bounds $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$. The projection of such a pair is given by the union of its dark shadow $(\alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) + (\alpha - 1)(\beta - 1) \leq 0)$ and another set of integer points which cannot be obtained by applying simple rules. The following theorem defines the hyperplanes on which the integer points lie.

LEMMA 3.5. *Let x, y be two rational numbers and $\lceil x \rceil, \lceil y \rceil$ (resp. $\lfloor x \rfloor, \lfloor y \rfloor$) be their upper (resp. lower) integer parts. The following properties are equivalent:*

- (1) $\exists n \in \mathbb{Z}$ such that $x \leq n \leq y$,
- (2) $\lceil x \rceil \leq \lfloor y \rfloor$,
- (3) $\lceil x \rceil \leq y$,
- (4) $x \leq \lfloor y \rfloor$.

THEOREM 3.6. *Consider the pair of bounds $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$ and let $l(\mathbf{x}, \mathbf{p}) = l_l(\mathbf{x}, \mathbf{p}) + c_l$ and $u(\mathbf{x}, \mathbf{p}) = l_u(\mathbf{x}, \mathbf{p}) + c_u$, where $l_l(\mathbf{x}, \mathbf{p})$ and $l_u(\mathbf{x}, \mathbf{p})$ are linear functions, c_l and c_u are integer constants. Let g be the greatest common divisor (gcd) of the coefficients of variables $\mathbf{x}$ and parameters $\mathbf{p}$ in the linear function $(\alpha l_l(\mathbf{x}, \mathbf{p}) - \beta l_u(\mathbf{x}, \mathbf{p}))$. Then*

—*the points outside the dark shadow which belong to the integer projection lie on hyperplanes of the form:*

$$\alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) + \gamma = 0, \quad (12)$$

with $\gamma \in \mathbb{Z}$, $0 \leq \gamma \leq \alpha\beta - \alpha - \beta$ and g divides $(\beta c_u - \alpha c_l - \gamma)$.

—*the values of γ for which the hyperplane (12) contains the points we are interested in are those verifying the following inequality:*

$$\alpha(-l(\mathbf{x}, \mathbf{p}) \bmod \beta) \leq \gamma, \quad (13)$$

which is equivalent to:

$$\beta(u(\mathbf{x}, \mathbf{p}) \bmod \alpha) \leq \gamma. \quad (14)$$

PROOF. We recall that the exact and the dark shadows are respectively given by $\alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) \leq 0$ and $\alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) \leq -(\alpha - 1)(\beta - 1)$ [Pugh and Wonnacott 1994]. By definition, the part of the exact shadow containing the points outside the dark shadow, which belong to the projection (see Figure 1), is given by $-(\alpha\beta - \alpha - \beta) \leq \alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) \leq 0$. This is equivalent to $\alpha l(\mathbf{x}, \mathbf{p}) - \beta u(\mathbf{x}, \mathbf{p}) + \gamma = 0$, where γ is an integer constant such that $0 \leq \gamma \leq \alpha\beta - \alpha - \beta$.

By substituting the values of $l(\mathbf{x}, \mathbf{p}) = l_l(\mathbf{x}, \mathbf{p}) + c_l$ and $u(\mathbf{x}, \mathbf{p}) = l_u(\mathbf{x}, \mathbf{p}) + c_u$ in (12) we obtain $\alpha l_l(\mathbf{x}, \mathbf{p}) - \beta l_u(\mathbf{x}, \mathbf{p}) = \beta c_u - \alpha c_l - \gamma$, where $(\alpha l_l(\mathbf{x}, \mathbf{p}) - \beta l_u(\mathbf{x}, \mathbf{p}))$ is a linear function and $(\beta c_u - \alpha c_l - \gamma)$ is an integer constant. A necessary and sufficient condition for this hyperplane to contain integer points is that the gcd of the coefficients in the linear function $(\alpha l_l(\mathbf{x}, \mathbf{p}) - \beta l_u(\mathbf{x}, \mathbf{p}))$ divides the constant $(\beta c_u - \alpha c_l - \gamma)$.

On the other hand, $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$ is equivalent to $\frac{l(\mathbf{x}, \mathbf{p})}{\beta} \leq z \leq \frac{u(\mathbf{x}, \mathbf{p})}{\alpha}$ (since $\alpha, \beta > 0$), where $\frac{l(\mathbf{x}, \mathbf{p})}{\beta}$ and $\frac{u(\mathbf{x}, \mathbf{p})}{\alpha}$ are rational functions. According to properties 1 and 2 of Lemma 3.5, there exists an integer z such that $\frac{l(\mathbf{x}, \mathbf{p})}{\beta} \leq z \leq \frac{u(\mathbf{x}, \mathbf{p})}{\alpha}$ if and only if:

$$\left\lceil \frac{l(\mathbf{x}, \mathbf{p})}{\beta} \right\rceil \leq \frac{u(\mathbf{x}, \mathbf{p})}{\alpha}, \quad (15)$$

with $\left\lceil \frac{l(\mathbf{x}, \mathbf{p})}{\beta} \right\rceil = \frac{1}{\beta}(l(\mathbf{x}, \mathbf{p}) + (-l(\mathbf{x}, \mathbf{p})) \bmod \beta)$. By simplifying (15), we obtain: $\alpha(-l(\mathbf{x}, \mathbf{p}) \bmod \beta) \leq \beta u(\mathbf{x}, \mathbf{p}) - \alpha l(\mathbf{x}, \mathbf{p})$, with $\beta u(\mathbf{x}, \mathbf{p}) - \alpha l(\mathbf{x}, \mathbf{p}) = \gamma$ (according to Equality (12)). Hence, inequality (13) is satisfied. Applying Lemma 3.5, we can similarly prove Inequality (14). $\square$

Note 3.7. If one of the bounds $l(\mathbf{x}, \mathbf{p})$ or $u(\mathbf{x}, \mathbf{p})$ is independent of the variables and the parameters, the elimination of the existential variable results in only one convex region, because in this case, $\lceil \frac{l(\mathbf{x}, \mathbf{p})}{\beta} \rceil$ or $\lfloor \frac{u(\mathbf{x}, \mathbf{p})}{\alpha} \rfloor$ is a constant.

Example 3.8. Consider the pair of bounds $\{7 \leq 4z, 3z \leq 2x + 5p + 1\}$. That is to say, $l(\mathbf{x}, \mathbf{p}) = 7$ (independent of the variables and parameters), $u(\mathbf{x}, \mathbf{p}) = 2x + 5p + 1$, $\alpha = 3$ and $\beta = 4$. The integer elimination of z results, applying inequality (15), in only one constraint: $\lceil \frac{7}{4} \rceil \leq \frac{2x+5p+1}{3}$, which is equivalent to $2x + 5p \geq 5$.

In the following, we describe how to calculate the solutions of inequalities (13) or (14), in two different ways, depending on whether the coefficients α and β are coprime or not.

3.3.1. Case of Coprime Coefficients.

THEOREM 3.9. Consider the pair of bounds $\{l(\mathbf{x}, \mathbf{p}) \leq \beta z, \alpha z \leq u(\mathbf{x}, \mathbf{p})\}$, with α, β coprime. The calculation of the values of γ for which hyperplane (12) contains the points, outside the dark shadow, and that belong to the integer projection does not depend on the variables and the parameters, that is, it depends only on the constants α and β . In this case, Inequalities (13) and (14) are respectively equivalent to (16) and (17).

$$\alpha((c_1\gamma) \bmod \beta) \leq \gamma, \quad (16)$$

$$\beta((c_2\gamma) \bmod \alpha) \leq \gamma, \quad (17)$$

where c_1 and c_2 are integer constants such that $c_1\alpha + c_2\beta = 1$ (computing these constants is straightforward from Bezout's identity theorem).

PROOF. Since α and β are coprime, there exists two integer constants c_1 and c_2 such that (Bezout's identity theorem):

$$c_1\alpha + c_2\beta = 1. \quad (18)$$

Multiplying Equality (12) by c_1 , we obtain $c_1\alpha l(\mathbf{x}, \mathbf{p}) - c_1\beta u(\mathbf{x}, \mathbf{p}) + c_1\gamma = 0$. This is equivalent to $(1 - c_2\beta)l(\mathbf{x}, \mathbf{p}) - c_1\beta u(\mathbf{x}, \mathbf{p}) + c_1\gamma = 0$ (according to (18)). Hence,

$$l(\mathbf{x}, \mathbf{p}) = \beta(c_2l(\mathbf{x}, \mathbf{p}) + c_1u(\mathbf{x}, \mathbf{p})) - c_1\gamma.$$

Substituting the value of $l(\mathbf{x}, \mathbf{p})$ in Inequality (13), we obtain:

$$\alpha((- \beta(c_2l(\mathbf{x}, \mathbf{p}) + c_1u(\mathbf{x}, \mathbf{p})) + c_1\gamma) \bmod \beta) \leq \gamma \Leftrightarrow \alpha(c_1\gamma \bmod \beta) \leq \gamma.$$

This proves Equation (16). In the same way, one can prove (17), starting from Inequality (14). $\square$

Example 3.10. Consider the following Presburger formula:

$$S = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} : 2 \leq 3y - x \leq 5 \wedge -1 \leq x - 2y \leq N - 1\},$$

where N is a positive integer parameter. This set is equivalent to the projection on x of the $\mathbb{Z}$ -polytope $\mathcal{Z}$ pictured in Figure 1 for $N = 2$.

According to the pair of bounds $\{x - N + 1 \leq 2y, 3y \leq x + 5\}$, we have:

$$l(x, N) = x - N + 1, \quad u(x, N) = x + 5, \quad \alpha = 3, \quad \beta = 2.$$

We can choose $c_1 = 1$ and $c_2 = -1$ ($c_1\alpha + c_2\beta = 1$).

The constraint on the dark shadow corresponding to this pair of bounds is $x \leq 3N + 5$. The points of the projection that do not belong to the dark shadow are

given by:

$$\begin{aligned} \alpha l(x, N) - \beta u(x, N) + \gamma &= 0, \quad 0 \leq \gamma \leq \alpha\beta - \alpha - \beta \text{ and } \alpha((c_1\gamma) \bmod \beta) \leq \gamma \\ \Rightarrow x - 3N - 7 + \gamma &= 0, \quad 0 \leq \gamma \leq 1 \text{ and } 3(\gamma \bmod 2) \leq \gamma. \end{aligned}$$

Scanning the different values of γ (ranging from 0 to the constant $\alpha\beta - \alpha - \beta$) we find that the only one satisfying these constraints is $\gamma = 0$. The corresponding hyperplane is $x = 3N + 7$. Similarly, one can calculate the point $x = 1$ from the other pair of bounds $\{x + 2 \leq 3y, 2y \leq x + 1\}$ generating the constraint $x \geq 3$ on the dark shadow.

The integer projection of $\mathbb{Z}$ -polytope $\mathcal{Z}$ is then obtained by intersecting the projections of both pairs, that is, $S = \{x = 3N + 7 \vee x \leq 3N + 5\} \cap \{x = 1 \vee x \geq 3\}$. Since N is positive, this set is equal to:

$$S = \{x \in \mathbb{Z} \mid x = 1 \vee 3 \leq x \leq 3N + 5 \vee x = 3N + 7\}.$$

3.3.2. Case of Non-Coprime Coefficients. In the previous section, we showed how we calculate the projection of a pair of bounds when the coefficients α and β are coprime. Let us now consider the case of non-coprime coefficients. In this case, the calculation of the values of γ for which the hyperplane (12) contains the points of the projection, and that are outside the dark shadow, depends not only on α and β , but also on the variables and the parameters. Let $g' = \gcd(\alpha, \beta)$, $\alpha' = \alpha/g'$, $\beta' = \beta/g'$ and g be the $\gcd$ of the coefficients of the variables and parameters in the linear function $\alpha' l_i(\mathbf{x}, \mathbf{p}) - \beta' l_u(\mathbf{x}, \mathbf{p})$, with $l(\mathbf{x}, \mathbf{p}) = l_l(\mathbf{x}, \mathbf{p}) + c_l$ and $u(\mathbf{x}, \mathbf{p}) = l_u(\mathbf{x}, \mathbf{p}) + c_u$ (see Theorem 3.6). One can then rewrite the equation of the hyperplane (12) as follows:

$$\alpha' l(\mathbf{x}, \mathbf{p}) - \beta' u(\mathbf{x}, \mathbf{p}) + \gamma = 0, \quad (19)$$

with $\gamma \in \mathbb{Z}$, $0 \leq \gamma \leq \alpha\beta' - \alpha' - \beta'$ and g divides $(\beta'c_u - \alpha'c_l - \gamma)$.

Inequalities (13) and (14) can be respectively rewritten as (20) and (21):

$$\alpha'(-l(\mathbf{x}, \mathbf{p}) \bmod \beta) \leq \gamma, \quad (20)$$

$$\beta'(u(\mathbf{x}, \mathbf{p}) \bmod \alpha) \leq \gamma. \quad (21)$$

In this case, only a sublattice of hyperplane (19) belongs to the projection since this one should also satisfy (20) or (21). These equations can be rewritten as:

$$-l(\mathbf{x}, \mathbf{p}) \bmod \beta = \gamma', \text{ with } 0 \leq \gamma' \leq \min\left(\left\lfloor \frac{\gamma}{\alpha'} \right\rfloor, \beta\right), \quad (22)$$

$$u(\mathbf{x}, \mathbf{p}) \bmod \alpha = \gamma', \text{ with } 0 \leq \gamma' \leq \min\left(\left\lfloor \frac{\gamma}{\beta'} \right\rfloor, \alpha\right). \quad (23)$$

In order to reduce the computation complexity, it is worth considering Equation (22) when $\beta < \alpha$ and Equation (23) otherwise: the smallest modulo will generate the less different cases.

The points verifying (20) or (21) lie on a union of lattices, since they verify Equality (22) or (23) for all possible values of γ' , and the solution to a *modulo* equality $f(\mathbf{x}, \mathbf{p}) \bmod a = b$ is a lattice. It has the form:

$$L = \left\{ (A_{\mathbf{x}} | A_{\mathbf{p}}) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \mathbf{c} \mid \mathbf{x} \in \mathbb{Z}^d, \mathbf{p} \in \mathbb{Z}^n \right\}, \quad (24)$$

where $A_{\mathbf{x}}, A_{\mathbf{p}}$ are integer matrices, $\mathbf{c}$ is an integer vector, $\mathbf{x}$ is a vector of the variables space and $\mathbf{p}$ is a vector of parameters. We calculate this solution using the technique presented in Section 3.1. Indeed, $f(\mathbf{x}, \mathbf{p}) \bmod a = b$ is equivalent to $\exists z \in \mathbb{Z} : f(\mathbf{x}, \mathbf{p}) = az + b$. It then suffices to calculate the validity lattice of this latter equation by eliminating existential variable z , or in other words, all integer values of $\mathbf{x}$

and $\mathbf{p}$ for which variable z is integer. Of course, only nonempty lattices and hyperplanes are taken into account.

Example 3.11. Consider a pair of bounds for which the coefficients of the existential variable y are not coprime $\{x - N - 2 \leq 2y, 4y \leq x + 5\}$. We have:

$$\begin{aligned} l(x, N) &= x - N - 2, \quad u(x, N) = x + 5, \quad \alpha = 4, \quad \beta = 2 \\ &\Rightarrow \gcd(\alpha, \beta) = 2, \quad \alpha' = 2, \quad \beta' = 1. \end{aligned}$$

The corresponding constraint on the dark shadow is $2x \leq 4N + 15 \Leftrightarrow x \leq 2N + 7$. The points outside the dark shadow and belonging to the projection lie on the following hyperplanes:

$$x - 2N - 9 + \gamma = 0, \text{ such that } 0 \leq \gamma \leq 1 \text{ and } 2(-(x - N - 2) \bmod 2) \leq \gamma.$$

For both values of γ , the solution to the above *modulo* inequality is a lattice:

$$L = \left\{ \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ N \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} \mid x \in \mathbb{Z}, N \in \mathbb{Z} \right\}.$$

Hence, points $x = 2N + 9$ and $x = 2N + 8$ (obtained by substituting the values of γ in equality $x - 2N - 9 + \gamma = 0$) belong to the projection only if x and N belong to lattice L .

The whole projection of the pair of bounds is then given by:

$$\mathcal{S} = \{x \in \mathbb{Z} \mid (x = 2N + 8 \wedge (x, N) \in L) \vee (x = 2N + 9 \wedge (x, N) \in L) \vee x \leq 2N + 7\}.$$

Projecting out a first existential variable from a $\mathbb{Z}$ -polytope may result in a union of *nonstandard* $\mathbb{Z}$ -polytopes. Therefore, in order to project out a second variable, this union has to be projected again, and so on. The projection of each nonstandard $\mathbb{Z}$ -polytope of the form $\mathcal{Z} = P \cap L$ is obtained as follows.

- Transform lattice L into an equivalent lattice L' in which the basis (linear part) is upper triangular.
- Compute polytope Q , the compression of $\mathcal{Z}$, which is the preimage of P by L' .
- Compute the projection P' of compressed polytope Q as explained before, and reduce lattice L' to L'' (remove the first row and the first column of L').
- Compute P'' the image of P' by L'' . The result of projecting $\mathcal{Z}$ is finally given by $\mathcal{Z}' = P'' \cap L''$.

The whole method of calculating the integer affine transformation of a $\mathbb{Z}$ -polytope is summarized in Algorithm 1 and 2.

3.3.3. Complexity Analysis. For eliminating a variable x , the maximum number of computed hyperplanes in the case of coprime coefficients is $(\sum_i |coef_i|)^2$, where the $coef_i$ are the coefficients of variable x appearing in the constraints, since this is the maximum number of values that all γ can take in Equation (12). In the case of non-coprime coefficients, it is multiplied by $\max_i (|coef_i|)$, the maximum number of lattices in Equation (22). In each case, we compute the intersection of an equality with a polytope, of complexity $nbCons$ (number of constraints) for fixed dimension. This procedure is repeated d' times, where d' is the number of eliminated variables. Finally, the worst-case complexity is:

$$\left(nbCons \left(\sum_i |coef_i| \right)^2 \max_i (|coef_i|) \right)^{d'}.$$

ALGORITHM 1: Calculating the integer affine transformation of a $\mathbb{Z}$ -polytope

```

Input:  $\mathcal{Z}$ :  $\mathbb{Z}$ -polytope;
         $T$ : Transformation matrix;
Output:  $\mathcal{Z}_u$ : Union of  $\mathbb{Z}$ -polytopes;
Data:  $F$ : Union of  $\mathbb{Z}$ -polytopes; // containing the successive projected sets
         $U$ : Union of  $\mathbb{Z}$ -polytopes; // initialized to Universe and successively intersected
        with the projection of pairs of bounds

// Preprocessing the transformation of  $\mathcal{Z}$  (Section 3.1, page 7)
 $F = \text{EliminateEqualities}(\text{PresburgerFormula}(\mathcal{Z}, T));$ 

// Remaining existential variables elimination
for each  $z$  in variables to be eliminated do
    // Transform  $\mathbb{Z}$ -polytopes  $F$  into standard polytopes (item 2, page 9)
     $F = \text{ZPolytopesCompression}(F);$ 
    //  $U = \mathbb{Z}^{\text{Dim}(F)-1}$ 
     $U = \text{Universe}(\text{Dimension}(F)-1);$ 
    for each  $(\alpha u, \beta l)$  in pairs of bounds on  $z$  in  $F$  do
        // Compute the Dark shadow (item 2, page 10)
         $D = \text{DarkShadow}(\alpha u, \beta l);$ 
        if  $\alpha == 1$  or  $\beta == 1$  then
             $U = U \cap D;$ 
        else
            // Compute the Exact shadow (item 1, page 10)
             $E = \text{ExactShadow}(\alpha u, \beta l);$ 
            // Compute and add hyperplanes from Theorem 3.6 (page 11)
            // function  $\text{ComputeHyperplanes}$  is detailed in Algorithm 2
             $U = U \cap (D \cup ((E - D) \cap \text{ComputeHyperplanes}(\alpha u, \beta l)))$ 
        end
    end
     $F = U;$ 
end
 $\mathcal{Z}_u = F;$ 

```

In many cases however, the execution time of this algorithm is reasonable, since this is a worst case complexity and we apply early simplifications at different points of the algorithm.

The result of Algorithm 1 is given as a union of parametric $\mathbb{Z}$ -polytopes. In the following section, we will be interested in counting points in such unions.

4. COUNTING POINTS IN UNIONS OF $\mathbb{Z}$ -POLYTOPES

We will now discuss an algorithm that deals with unions of a fixed number of parametric $\mathbb{Z}$ -polytopes of the form $\mathcal{Z} = P \cap L$, with:

$$P = \left\{ \mathbf{x} \in \mathbb{Q}^d, \mathbf{p} \in \mathbb{Z}^n \mid (A_{\mathbf{x}} \mid A_{\mathbf{p}}) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \mathbf{a} \geq 0 \right\},$$

$$L = \left\{ (B_{\mathbf{x}} \mid B_{\mathbf{p}}) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \mathbf{b} \mid \mathbf{x} \in \mathbb{Z}^d, \mathbf{p} \in \mathbb{Z}^n \right\},$$

where P is a parametric polytope, L is a parametric integer lattice, $A_{\mathbf{x}}$, $A_{\mathbf{p}}$, $B_{\mathbf{x}}$ and $B_{\mathbf{p}}$ are integer matrices, $\mathbf{a}$ and $\mathbf{b}$ are integer vectors, $\mathbf{x}$ is a vector of the variables space and $\mathbf{p}$ is a vector of parameters.

Let $\mathcal{Z}_1 = P_1 \cap L_1$ and $\mathcal{Z}_2 = P_2 \cap L_2$ be two $\mathbb{Z}$ -polytopes. The number of points of $\mathcal{Z}_1$ is equal to the number of integer points contained in P'_1 , where P'_1 is the transformation of P_1 by the matrix B_1 defining lattice L_1 . In the same way, the number of

ALGORITHM 2: Function ComputeHyperplanes (called by Algorithm 1)

Input: $(\alpha u, \beta l)$: pair of bounds;
Output: H_u : Union of $\mathbb{Z}$ -polyhedra; // union of sublattices of hyperplanes
Data: L_u : Union of lattices; // solutions to modulo inequalities

$H_u = \text{Empty};$
if Coprime (α, β) **then**
 $(c_1, c_2) = \text{Bezout}(\alpha, \beta);$ // such that $\alpha c_1 + \beta c_2 = 1$ by Bezout's theorem
 for $\gamma = 0$ **to** $\alpha\beta - \alpha - \beta$ **do**
 if $\alpha((c_1\gamma) \bmod \beta) \leq \gamma$ **then**
 $H_u = H_u \cup \{\alpha l - \beta u + \gamma = 0\};$
 end
 end
else
 $\alpha' = \alpha / \gcd(\alpha, \beta);$
 $\beta' = \beta / \gcd(\alpha, \beta);$
 if $\beta < \alpha$ **then**
 for $\gamma = 0$ **to** $\alpha\beta' - \alpha' - \beta'$ **do**
 $L_u = \text{Solve}(\alpha'(-l \bmod \beta) \leq \gamma);$ // complexity depends on β
 $H_u = H_u \cup (\{\alpha'l - \beta'u + \gamma = 0\} \cap L_u);$
 end
 else
 for $\gamma = 0$ **to** $\alpha\beta' - \alpha' - \beta'$ **do**
 $L_u = \text{Solve}(\beta'(u \bmod \alpha) \leq \gamma);$ // complexity depends on α
 $H_u = H_u \cup (\{\alpha'l - \beta'u + \gamma = 0\} \cap L_u);$
 end
 end
end
end

Fig. 2. Union of two $\mathbb{Z}$ -polytopes.

points of $\mathcal{Z}_2$ is equal to the number of integer points in $P'_2 = B_2 P_2$. Unfortunately, the number of points in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is obviously not equal to that in $P'_1 \cup P'_2$ since the applied transformations preserve the number of points in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ but do not preserve their original coordinates.

Example 4.1. Consider the two $\mathbb{Z}$ -polytopes pictured in Figure 2, $\mathcal{Z}_1 = P_1 \cap L_1$ and $\mathcal{Z}_2 = P_2 \cap L_2$, where dots belong to $\mathcal{Z}_1$, squares belong to $\mathcal{Z}_2$ and diamonds belong to

both $\mathbb{Z}$ -polytopes:

$$\begin{aligned} P_1 &= \{(x, y) \in \mathbb{Q}^2 \mid 1 \leq x \leq 10 \wedge 3 \leq y \leq 7\}, \\ L_1 &= \left\{ \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \mid (x', y') \in \mathbb{Z}^2 \right\}, \\ P_2 &= \{(x, y) \in \mathbb{Q}^2 \mid 3 \leq x \leq 12 \wedge 1 \leq y \leq 6\}, \\ L_2 &= \left\{ \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \mid (x', y') \in \mathbb{Z}^2 \right\}. \end{aligned}$$

Substituting $x = 2x' + 1$ and $y = 2y' + 1$ in P_1 (resp. $x = 3x'$ and $y = 2y' + 1$ in P_2) we obtain P'_1 and P'_2 in which the numbers of points are respectively 15 and 12:

$$\begin{aligned} P'_1 &= \{(x', y') \in \mathbb{Z}^2 \mid 0 \leq 2x' \leq 9 \wedge 1 \leq y' \leq 3\}, \\ P'_2 &= \{(x', y') \in \mathbb{Z}^2 \mid 1 \leq x' \leq 4 \wedge 0 \leq 2y' \leq 5\}. \end{aligned}$$

One can check that the number of points in $P'_1 \cup P'_2$ is 19, whereas that in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is 23 as shown in Figure 2.

The previous methods [Meister 2004; Zhu et al. 2006] to count points in unions of $\mathbb{Z}$ -polytopes are *lattice-union* based, which is exponential in the size of lattice generators and their least common multiple. Furthermore, Zhu et al.'s method [2006] deals only with nonparametric $\mathbb{Z}$ -polytopes.

In contrast, our method is *lattice-intersection* based: its complexity is polynomial (for a fixed number of $\mathbb{Z}$ -polytopes), since the intersection between two lattices results in only one lattice, whatever their generators. Previous algorithms start by calculating a disjoint union of the input $\mathbb{Z}$ -polytopes. It is usually very hard to separate a union of $\mathbb{Z}$ -polytopes into a *disjoint* union [Zhu et al. 2006], and its complexity may be exponential even for a fixed number of $\mathbb{Z}$ -polytopes.

In Algorithm 3, we start by applying the inclusion-exclusion principle to the union of $\mathbb{Z}$ -polytopes. We therefore process a set of signed $\mathbb{Z}$ -polytopes: $\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2)$, the number of integer points in the union of two $\mathbb{Z}$ -polytopes $\mathcal{Z}_1$ and $\mathcal{Z}_2$, is equal to $\mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_1 \cap \mathcal{Z}_2)$.

After applying the inclusion-exclusion principle, the number of points in each resulting $\mathbb{Z}$ -polytope $\mathcal{Y}_i = P_i \cap L_i$ is calculated as follows:

We first transform the matrix generating the lattice:

$$L_i = \left\{ (B_{\mathbf{x}} \mid B_{\mathbf{p}}) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \begin{pmatrix} \mathbf{b}_{\mathbf{x}} \\ \mathbf{b}_{\mathbf{p}} \end{pmatrix} \mid \mathbf{x} \in \mathbb{Z}^d, \mathbf{p} \in \mathbb{Z}^n \right\} \text{ into a new matrix of the form: } M = \begin{pmatrix} B_{\mathbf{xx}} & B_{\mathbf{xp}} \\ 0 & B_{\mathbf{pp}} \end{pmatrix}.$$

Matrix M generates the same integer points as the original matrix, and the rows of M defining the lattice in the parameter space are independent of the variables. This is required to keep the variables space compressed when rewriting the transformed $\mathbb{Z}$ -polytope as a function of its original parameters. We calculate matrix M from the Hermite normal form [Schrijver 1986] of matrix $(B_{\mathbf{x}} \mid B_{\mathbf{p}})$, with rows and columns exchanged such that parameters appear first. This is needed to obtain zeroes under $B_{\mathbf{xx}}$ in M . We apply the affine transformation $\left(M \mid \begin{pmatrix} \mathbf{b}_{\mathbf{x}} \\ \mathbf{b}_{\mathbf{p}} \end{pmatrix} \right)$ to P_i to get a polytope P' . Then, P' is rewritten as a function of the original parameters using the submatrix $(B_{\mathbf{pp}} \mid \mathbf{b}_{\mathbf{p}})$ and finally, we use our counting algorithm [Verdoolaege et al. 2004, 2007] to calculate the Ehrhart quasi-polynomial corresponding to the number of integer points in the resulting polytope. Note that when submatrix $B_{\mathbf{pp}}$ is not equal to the identity matrix, the polytope is valid for only the parameter values generated by lattice $L_{\mathbf{p}}$ whose basis is $B_{\mathbf{pp}}$ and affine part is $\mathbf{b}_{\mathbf{p}}$. In this case, the resulting Ehrhart quasi-polynomial is to be multiplied by one if the parameter values are valid (i.e., if they belong to the points spanned by lattice $L_{\mathbf{p}}$) and zero otherwise.

ALGORITHM 3: Counting integer points in a union of parametric $\mathbb{Z}$ -polytopes

Input: F : Union of $\mathbb{Z}$ -polytopes;
Output: L : List of (Chamber, Ehrhart quasi-polynomial);
Data: S, S' : List of (sign, $\mathbb{Z}$ -polytope);
 P : Polytope; I : $\mathbb{Z}$ -polytope;
 L' : List of (Chamber, Ehrhart quasi-polynomial);

```

// Apply inclusion-exclusion principle to the set  $F$ 
 $S = \text{Empty}$ ;
for each  $Z$  in  $F$  do
   $S' = S$ ;
  for each  $(s, \mathcal{Y})$  in  $S$  do
     $I = Z \cap \mathcal{Y}$ ;
    if Not Empty ( $I$ ) then
       $S' = S' + (-1 \times s, I)$ ;
    end
  end
   $S = S' + (+1, Z)$ ;
end

// Compute the number of integer points in the set  $S$ 
 $L = \text{Empty}()$ ;
for each  $(s, \mathcal{Y})$  in  $S$  do
  // Compute an equivalent lattice to  $\mathcal{Y}.L$  in which the parameter rows are
  // independent of the variables (matrix  $M$ , page 18)
   $\mathcal{Y}.L = \text{NormalizeLattice}(\mathcal{Y}.L)$ ;
  // Compute the transformation of polytope  $\mathcal{Y}.P$  using the normalized lattice
  // (polytope  $P'$ , page 18)
   $P = \text{Transform}(\mathcal{Y}.P, \mathcal{Y}.L)$ ;
  // Count integer points in  $P$  (using [Verdoolaege et al. 2007])
   $L' = s \times \text{Enumerate}(P)$ ;
  // Add  $L'$  to  $L$  and simplify the result (Note 4.2, page 19)
   $L = \text{AddAndSimplify}(L, L')$ ;
end

```

Note 4.2. The results coming from the different $\mathbb{Z}$ -polytopes (lists of pairs³ (D, E)) have to be combined into a single list. We do this as follows:

Two lists L_1 and L_2 , each one containing one element, are combined into one list as: $(L_1.D \setminus L_2.D, L_1.E), (L_2.D \setminus L_1.D, L_2.E)$ and $(L_1.D \cap L_2.D, L_1.E + L_2.E)$.

The generalization of this process for many lists containing many elements is straightforward, but the result usually has to be simplified in order to reduce its size.

Example 4.3. Consider two $\mathbb{Z}$ -polytopes $Z_1 = P_1 \cap L_1$ and $Z_2 = P_2 \cap L_2$ (derived from Example 4.1 by adding a parameter N), with

$$P_1 = \{(x, y) \in \mathbb{Q}^2 \mid 1 \leq x \leq N + 5 \wedge 3 \leq y \leq 7\},$$

$$L_1 = \left\{ \left(\begin{array}{cc|c} 2 & 0 & 3 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array} \right) \begin{pmatrix} x' \\ y' \\ N' \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \mid (x', y') \in \mathbb{Z}^2, N' \in \mathbb{Z} \right\},$$

³ D = chamber and E = Ehrhart polynomial.

$$P_2 = \{(x, y) \in \mathbb{Q}^2 \mid 3 \leq x \leq 2N + 7 \wedge 1 \leq y \leq 6\},$$

$$L_2 = \left\{ \left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{array} \right) \begin{pmatrix} x' \\ y' \\ N' \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \mid (x', y') \in \mathbb{Z}^2, N' \in \mathbb{Z} \right\},$$

where $N \in \mathbb{Z}^+$ is a parameter. The number of points in the union $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is given by:

$$\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2) = \mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_1 \cap \mathcal{Z}_2).$$

Let $(\mathcal{Z}_1 \cap \mathcal{Z}_2) = \mathcal{Z}_3 = (P_3 \cap L_3)$, with

$$P_3 = P_1 \cap P_2 = \{(x, y) \in \mathbb{Q}^2 \mid 3 \leq x \leq N + 5 \wedge 3 \leq y \leq 6\},$$

$$L_3 = L_1 \cap L_2 = \left\{ \left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 3 & 0 & 6 \end{array} \right) \begin{pmatrix} x' \\ y' \\ N' \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} \mid (x', y') \in \mathbb{Z}^2, N' \in \mathbb{Z} \right\}.$$

The number of integer points in $\mathcal{Z}_3$ is calculated as follows.

First of all, basis $\left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 3 & 0 & 6 \end{array} \right)$ of lattice L_3 is transformed into a new basis $M = \left(\begin{array}{ccc|c} 6 & 0 & 3 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 3 & 3 \end{array} \right)$ in which the variable coefficients in the parameter row are all equal to zero. The new basis spans the same integer points as the original one since it is calculated from its Hermite normal form. $\mathbb{Z}$ -polytope $\mathcal{Z}_3$ is then transformed into a regular polytope P given by the preimage of P_3 by homogeneous matrix $\left(\begin{array}{ccc|c} 6 & 0 & 3 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 \end{array} \right)$:

$$P = \{(x', y') \in \mathbb{Q}^2 \mid -3N' + 3 \leq 6x' \leq 8 \wedge 2 \leq 2y' \leq 5\}.$$

Before counting points in P , we need to write it as a function of the original parameter N . To do this, it suffices to calculate its transformation by matrix $M = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 \end{array} \right)$, and we obtain:

$$P' = \{(x', y') \in \mathbb{Q}^2 \mid -N + 6 \leq 6x' \leq 8 \wedge 2 \leq 2y' \leq 5\}.$$

The number of integer points in this polytope is given (using the algorithm described in [Verdoolaege et al. 2004, 2007]) by:

$$\mathcal{E}(P') = \frac{1}{3}N + \left[2, \frac{5}{3}, \frac{4}{3}, 1, \frac{2}{3}, \frac{1}{3} \right]_N,$$

where $\left[2, \frac{5}{3}, \frac{4}{3}, 1, \frac{2}{3}, \frac{1}{3} \right]_N$ is a periodic function whose value is 2 when $N \bmod 6 = 0$, $\frac{5}{3}$ when $N \bmod 6 = 1$ and so on.

The third row in matrix M states that $N = 3N' + 3$, with $N' \in \mathbb{Z}$. In other words, N must be a multiple of 3 ($N \bmod 3 = 0$). Therefore the resulting polynomial is multiplied by a periodic function $[1, 0, 0]_N$ which is equal to one when $N \bmod 3 = 0$ and zero otherwise. The result is then:

$$\mathcal{E}(\mathcal{Z}_3) = \left[\frac{1}{3}, 0, 0 \right]_N N + [2, 0, 0, 1, 0, 0]_N$$

The numbers of points in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are obtained in a similar way as:

$$\mathcal{E}(\mathcal{Z}_1) = \frac{3}{2}N + \left[9, \frac{15}{2} \right]_N,$$

$$\mathcal{E}(\mathcal{Z}_2) = [2, 0, 0]_N N + [6, 0, 0]_N.$$

Finally, the number of points in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is:

$$\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2) = \mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_3) = \left[\frac{19}{6}, \frac{3}{2}, \frac{3}{2} \right]_N N + \left[13, \frac{15}{2}, 9, \frac{25}{2}, 9, \frac{15}{2} \right]_N.$$

Complexity analysis. We will discuss the complexity of our algorithm when the dimension and the number of input $\mathbb{Z}$ -polytopes is fixed.

The complexity of Algorithm 3 depends on the complexity of the significant polytope and $\mathbb{Z}$ -polytope operations, the complexity of counting integer points in a parametric polytope and the number of the resulting $\mathbb{Z}$ -polytopes after applying the inclusion-exclusion principle.

- The polytope operations rely on PolyLib, which is based on the dual *rays* and *constraints* representations. All the operations that we use, namely affine transformation and intersection, are polynomial for fixed dimension. The theoretical maximum complexity of the constraints to rays computation is $c^{\lfloor \frac{d}{2} \rfloor}$, where c is the number of constraints and d the dimension of the polytope.
- The only significant $\mathbb{Z}$ -polytope operation used in this algorithm is the intersection, which is polynomial since the intersection between two polytopes is a concatenation of their constraints, and the intersection between two lattices reduces to solving a system of linear equalities [Nookala and Risset 2000], which is polynomial in the input size [Schrijver 1986].
- Counting integer points in a parametric polytope is also polynomial in the input size (for fixed dimension), as we showed in [Verdoolaege et al. 2004, 2007].
- Finally, the $\mathbb{Z}$ -polytopes to be enumerated are given by the inclusion-exclusion principle. When the number of input $\mathbb{Z}$ -polytopes is fixed, the inclusion-exclusion principle provides a polynomial number of (nonempty) $\mathbb{Z}$ -polytopes. The theoretical complexity of this algorithm is $2^n - 1$, where n is the number of input $\mathbb{Z}$ -polytopes.

Hence, the whole algorithm is polynomial in the input size for fixed dimension and fixed number of input $\mathbb{Z}$ -polytopes, but, of course, exponential for arbitrary dimension and arbitrary number of input $\mathbb{Z}$ -polytopes.

5. EXPERIENCES, RELATED WORK, AND APPLICATIONS

In this section, we compare our work to some recent and interesting methods over a set of problems that frequently arise in code optimization techniques. All the execution times that are given were measured on a 3GHz Intel Pentium 4 with 1GB of RAM running Linux 2.6.23.

5.1. Omega Test

We first illustrate the difference between our method and Pugh et al.'s Omega test [Pugh 1991, 1994; Pugh and Wonnacott 1994]. The Omega test answers the question: *is there an integer point in the integer projection of a polytope?* as follows: it first computes the *exact shadow* (rational projection). If it does not contain any integer point, then the answer is “no”. Then it computes the *dark shadow*. If it contains at least an integer point, then the answer is “yes”; else Pugh and Wonnacott [1994] propose to check whether the intersection between a certain number of hyperplanes and the constraints of the original system contains an integer point. Their solution provides new constraints with possibly extra existential variables (the *splinters*) which increases the complexity of further computations, such as counting the number of integer points contained in the integer projection of the $\mathbb{Z}$ -polytope, or projecting the result along another dimension.

Note that the splinters provided by Pugh's method are somewhat similar to our $\mathbb{Z}$ -polytopes (when the coefficients of the existential variable are not coprime, see Section 3.3.2). But it is not clear whether these splinters can be projected along another dimension without scanning a possibly exponentially growing number of $\mathbb{Z}$ -polytopes. Also, when the coefficients are coprime, Pugh's method does not propose a simple solution as we do, and no explanation is given for non-coprime coefficients nor for the parametric case.

Example 5.1. Consider the following example (introduced in Pugh and Wonnacott [1994]):

$$S = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} : 0 \leq 3y - x \leq 7 \wedge 1 \leq x - 2y \leq 5\}.$$

The exact shadow defined by the elimination of y is given by $3 \leq x \leq 29$ and the dark shadow by $5 \leq x \leq 27$.

Pugh and Wonnacott [1994] calculate the set of constraints containing the images which do not belong to the dark shadow as follows:

$$\begin{aligned} & \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} : x = 3y \wedge 1 \leq y \leq 5\} \\ & \cup \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} : x = 3y - 1 \wedge 2 \leq y \leq 6\} \\ & \cup \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} : x = 2y + 5 \wedge 5 \leq y \leq 12\}, \end{aligned}$$

which has to be intersected with the difference between the exact shadow and the dark shadow in order to obtain the solution. While our rules provide these images directly as $\{x = 3, x = 29\}$.

5.2. Gupta and Rajopadhye's $\mathbb{Z}$ -Polyhedral Model

In 2007, Gupta and Rajopadhye presented new approaches for manipulating $\mathbb{Z}$ -polyhedra [Gupta and Rajopadhye 2007]. In particular, they prove closure in the $\mathbb{Z}$ -polyhedral model under integer affine images: the integer affine image of a union of $\mathbb{Z}$ -polyhedra is a finite union of $\mathbb{Z}$ -polyhedra, but the number of $\mathbb{Z}$ -polyhedra of the union may be exponential in the input size.

They propose a complete algorithm to compute such an image. Basically, it consists in computing a union of subsets of a $\mathbb{Z}$ -polyhedron by translating it, and then to cut each $\mathbb{Z}$ -polyhedron of the resulting union with hyperplanes. The number of those cutting hyperplanes, each one resulting in a union of $\mathbb{Z}$ -polyhedra of the final computed union, may be exponential in the input size.

Let us consider Example 4 from Gupta and Rajopadhye [2007]. Let $P = \{(i, j) \in \mathbb{Z}^2 \mid i \geq 0 \wedge j \geq 0 \wedge i + j \leq 3\}$ be a $\mathbb{Z}$ -polyhedron, and $T(i, j) = i + 3j$ a transformation. The computation of the image of P by T is given by their algorithm as:

$$\{i \mid 0 \leq i \leq 3, i \in \mathbb{Z}\} \cup \{2i + 3 \mid 0 \leq i \leq 3, i \in \mathbb{Z}\} \cup \{2i + 2 \mid 0 \leq i \leq 2, i \in \mathbb{Z}\}$$

while we compute directly, thanks to early simplifications (like considering the dark shadow or coprime coefficients):

$$\{i \in \mathbb{Z} \mid 0 \leq i \leq 7\} \cup \{i = 9\}.$$

The first $\mathbb{Z}$ -polytope that we compute here is the dark shadow.

We plan to perform a more accurate comparison of both approaches, on a wide set of test examples, as soon as their algorithm will be implemented.

5.3. Weak Quantifier Elimination

A recent and interesting work manipulating Presburger formulas is the one due to Lasaruk and Sturm [2007]. It consists in a generalization of Presburger arithmetic, where the coefficients are arbitrary polynomials in nonquantified variables. The authors use the term "weak" to refer to the fact that the result may still contain bounded

quantifiers. In order to expand this result into disjunctions, the parameters have to be instantiated. In the following, we will show that, at least for a subset of problems⁴, our method also handles parameters and can do better, in particular in terms of execution time.

Let us consider an interesting issue raised by the code optimization community, which is dependency analysis for automatic parallelization. More precisely, we consider the example from Lasaruk and Sturm [2007] (Section 5.2). The goal is to check whether a data dependency occurs or not in the following loop nest:

```
for i = 0 to m do
  for j = 0 to m do
    A[n*i+j] = i+j
```

There is a data dependency when two or more different loop iterations write different values to a same array element. For this loop nest, this condition can be expressed as the following Presburger formula:

$$\exists i \exists j \exists i' \exists j' (0 \leq i \leq m \wedge 0 \leq j \leq m \wedge 0 \leq i' \leq m \wedge 0 \leq j' \leq m \\ \wedge (i \neq i' \vee j \neq j') \wedge i + j \neq i' + j' \wedge ni + j = ni' + j').$$

Lasaruk and Sturm [2007] compute in 3.53s that the weak quantifier elimination over integers from the above formula results in 25441 atomic formulas, which cannot be expanded unless the parameters are substituted. They also raise the fact that it is more efficient to first plug in values for the parameters m and n , and then to perform quantifier elimination plus expansion. When they do so, they obtain “true” for $m = n = 4$ in 0.28s and “false” for $m = 4$ and $n = 5$ in 0.29s.

Let us now see how this problem can efficiently be solved using our method:

When $m = n = 4$ we obtain “yes” in 0.01s, and we obtain “false” when $m = 4$ and $n = 5$ in less than 0.01s.

Better, our method has only to plug in the value of parameter n (since we do not handle polynomial constraints). In this case, we obtain:

- when $n = 4$: “true” if $m \geq 4$ and “false” otherwise in less than 0.01s,
- when $n = 5$: “true” if $m \geq 5$ and “false” otherwise in 0.01s.

Notice that the authors do not mention what could be the result in this second case.

Another information that can be given by our method is the array elements causing the data dependency and their count. This is done by replacing equality $ni + j = ni' + j'$ by $x = ni + j \wedge x = ni' + j'$ in the previous formula, where x is a free-quantifier variable representing accessed array elements. Again, only parameter n has to be substituted. For example, when $n = 4$, the number of array cells causing a dependency is given as: {4 if $m = 4$, 10 if $m = 5$, 18 if $m = 6$ and $5m - 7$ if $m \geq 7$ }.

5.4. Barvinok-Woods’s Integer Projection

Barvinok and Woods [2003] proposed an algorithm for computing the integer projection of a polytope that is *theoretically* polynomial-time for fixed dimension. However, no implementation of this algorithm has been reported till 2008. Because it contains complicated theory such as algorithmic flatness theory and iterated Boolean combinations of rational generating functions, this algorithm has frequently been referred to as “unimplementable.” Köppe et al. [2008] rose to the challenge of proposing the first-ever implementation of Barvinok-Woods’s algorithm. In their paper, the authors present the

⁴Our method handles only linear constraints (polynomial constraints are not supported).

Table II. Köppe et al.'s Implementation Execution Times

Example	<i>ex1</i>	<i>woods 2.1.7</i>	<i>pugh</i>	<i>scarf.1</i>
# parameters	0	1	0	2
# free-quantifier variables	0	0	0	0
# existentially quantified variables	2	2	2	2
# inequalities	4	4	4	5
<i>Köppe's time without exploiting small gaps in dimension 2</i>	0.11	29.2	0.09	126
<i>Köppe's time exploiting small gaps in dimension 2</i>	0.08	2.7	?	1.1

results of preliminary computational experiments. Unfortunately, their implementation turned out to be rather inefficient, as shown in the experiments they reported, in particular *woods 2.1.7* and *scarf.1*. Table II summarizes execution times (in seconds) obtained by Köppe et al.'s implementation.

For all these examples, our method computes the result in less than 0.01s. Notice that we did not test our method on the *param.pugh* example, because the authors did not make it public.

5.5. Verdoolaege's Integer Projection

In this section, we compare our implementation with the one written by Sven Verdoolaege and distributed in the Barvinok library [Verdoolaege 2006]. This implementation consists in applying simple rewriting rules to a set of disjoint union of parametric sets computed using the Omega library [Kelly et al. 1996]. If these rules fail, the PIP library [Feautrier et al. 2002] is used to solve a parametric integer linear programming problem, in the way of Boulet and Redon [1998]. Notice that both tools (Omega and PIP) are worst-case exponential.

The particular advantage of our method over the one of Verdoolaege is its ability to handle efficiently *unions* of projections. Indeed, in such a case, our algorithm computes each projection independently of the others, even if the original sets are not pairwise disjoint projections. The final result is then easily obtained by doing the union of the resulting atomic projections. The number of lattice points contained in this union is computed straightforwardly using the algorithm proposed in Section 4. Verdoolaege's method could not do so since, in contrast to our method, it does not compute the actual atomic projections but equivalent sets having the same number of lattice points. Hence, the original input sets have to be pairwise disjoint. Furthermore, the equivalent resulting sets are usually of higher dimension than the actual projections. This, usually, leads to higher lattice points counting times and results in larger Ehrhart quasi-polynomials.

Example 5.2. Consider the following two loop nests accessing a one-dimensional array:

```

for i = 1 to n do
  for j = i+1 to n do
    A[2*i+3*j] = ...

for k = 1 to n do
  for l = 1 to k-1 do
    A[k+2*l] = ...

```

Suppose we are interested in the number of array elements that are reached by at least one of these loop nests iterations. This is useful, for example, to compute the amount of data being accessed by this piece of program (the cache size used by this computation could also be computed). This in turn reduces to the problem of counting the solutions

to the following Presburger formula:

$$\begin{aligned} \exists i \exists j (1 \leq i \leq n \wedge i + 1 \leq j \leq n \wedge x = 2i + 3j) \\ \vee \exists k \exists l (1 \leq k \leq n \wedge 1 \leq l \leq k - 1 \wedge x = k + 2l). \end{aligned}$$

Verdoolaege's algorithm first uses the Omega library to compute three disjoint formulas which still contain existential variables, and then applies the rewriting rules and/or the PIP library to compute their integer projections, which results in a set of disjoint polytopes. Applying the lattice points counting algorithm to this set results in 3-degree Ehrhart quasi-polynomials. The solution is given in 0.12s computation time and the size of the output file is 2475 bytes.

In contrast to this, our method computes the result in only 0.01s and the output file takes 256 bytes only. The result we obtain is: 2, 6, 11, 16, 22 when n respectively goes from 2 to 6, and $5n - 7$ when n is greater or equal to 7. Notice that the size of the output file has to be as small as possible, since it contains Ehrhart quasi-polynomials that are sometimes used in optimizing code applications.

In order to measure the efficiency of our method, we also ran a simulation through a thousand of pseudo-randomly generated examples that model array accesses in perfect loop nests. The examples are constructed as follows.

- The depth of loop nests (number of the existential variables) varies from 1 to 6.
- The dimension of arrays (number of free-quantifier variables) varies from 1 to 4.
- The number of parameters varies from 1 to 4.
- The number of equalities equals the array dimension.
- The number of inequalities equals 2 times the depth of the loop nest.
- The coefficients of the variables and parameters are chosen such that they reflect what could be found in a real program.

Consequently, the number of dimensions of the generated sets⁵ varies from 3 to 14, and the number of constraints varies from 3 to 16.

In these experiments, our implementation and Verdoolaege's compute the solution in 0.01s or less for 804 examples. For 49 examples, they did not compute in the allocated time that we fixed to a couple of minutes. For the remaining examples, we get a mean acceleration of 1.64 compared to Verdoolaege's implementation. Figure 3 shows the acceleration that we obtain for each example, where 13 circled points at the top of the diagram (respectively 7 at the bottom) are the examples for which Verdoolaege's (resp. our) implementation did not finish in the allocated time.

Notice that, even if all the examples in this simulation are the image of a single polytope (we did not compute the image of a union of $\mathbb{Z}$ -polytopes), our method yields to better results than the one of Verdoolaege.

This simulation was undertaken with the PolyLib library version 5.23 and the Barvinok library version 0.28. In both implementations, PolyLib is used to realize polyhedral operations, and the Barvinok library to count integer points in parametric polytopes. In addition, Verdoolaege uses the PIP library [Feautrier et al. 2002] and the Omega library version 2.1 [Kelly et al. 1996] to simplify the input polytopes.

6. CONCLUSION

We presented a new algorithm for calculating the integer affine transformation of parametric $\mathbb{Z}$ -polytopes. The solution is given as a union of parametric $\mathbb{Z}$ -polytopes, worst-case exponential but efficient in practical cases, and less complex compared to other existing methods. A general polynomial algorithm for computing the exact

⁵We consider only nonempty sets.

Fig. 3. Execution time comparison with Verdoolae's implementation.

solution remains a challenge. We also proposed a new polynomial algorithm to count points in arbitrary unions of a fixed number of $\mathbb{Z}$ -polytopes (of fixed dimension).

These algorithms have been implemented⁶ using the PolyLib library [Loechner 1999] and the Barvinok library [Verdoolae et al. 2004, 2007]. They have many applications, particularly in compiler design (for example analyzing and optimizing loop nests), and in other domains such as economics and mathematics (for example in combinatorics, representation theory, statistics, discrete optimization).

ACKNOWLEDGMENT

We would like to thank Sanjay Rajopadhye and the anonymous reviewers for their insights and valuable feedback.

REFERENCES

- ANCOURT, C., COELHO, F., IRIGOIN, F., AND KERYELL, R. 1997. A linear algebra framework for static HPF code distribution. *Scient. Program.* 6, 1, 3–27.
- BALDONI-SILVA, M. W., BECK, M., COCHET, C., AND VERGNE, M. 2006. Volume computation for polytopes and partition functions for classical root systems. *Discr. Comput. Geom.* 35, 4, 551–595.
- BARVINOK, A. AND WOODS, K. 2003. Short rational generating functions for lattice point problems. *J. Amer. Math. Soci.* 16, 657–979.
- BARVINOK, A. I. 1994a. Computing the Ehrhart polynomial of a convex lattice polytope. *Discrete Comput. Geom.* 12, 35–48.
- BARVINOK, A. I. 1994b. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.* 19, 4, 769–779.
- BOIGELOT, B. AND LATOUR, L. 2004. Counting the solutions of Presburger equations without enumerating them. *Theor. Comput. Sci.* 313, 1, 17–29.
- BOULET, P. AND REDON, X. 1998. Communication pre-evaluation in HPF. In *Proceedings of the 4th International Euro-Par Conference on Parallel Processing*. Lecture Notes in Computer Science, vol. 1470, Springer, 263–272.
- CLAUSS, P. AND LOECHNER, V. 1998. Parametric analysis of polyhedral iteration spaces. *J. VLSI Signal Process.* 19, 2, 179–194.

⁶Available under GNU Public Licence on <http://ZPolyTrans.gforge.inria.fr>.

- CREUSILLET, B. AND IRIGOIN, F. 1996. Interprocedural array region analyses. *Int. J. Parallel Program.* 24, 513–546.
- D'ALBERTO, P., VEIDEMBAUM, A., NICOLAU, A., AND GUPTA, R. 2001. Static analysis of parameterized loop nests for energy efficient use of data caches. In *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP '01)*.
- DANTZIG, G. B. AND EAVES, B. C. 1973. Fourier-Motzkin elimination and its dual. *J. Comb. Theory, Ser. A* 14, 3, 288–297.
- DE LOERA, J. A., HEMMECKE, R., TAUZER, J., AND YOSHIDA, R. 2004. Effective lattice point counting in rational convex polytopes. *J. Symbol. Comput.* 38, 4, 1273–1302.
- EHRHART, E. 1977. *Polynômes Arithmétiques et Méthode des Polyèdres en Combinatoire*. International Series of Numerical Mathematics 35.
- FEAUTRIER, P. 1988. Parametric integer programming. *Recherche Operationnelle / Operations Research* 22, 3, 243–268.
- FEAUTRIER, P., COLLARD, J., AND BASTOUL, C. 2002. Solving systems of affine (in)equalities. Tech. rep., PRISM, Versailles University.
- GHOSH, S., MARTONOSI, M., AND MALIK, S. 1999. Cache miss equations: a compiler framework for analyzing and tuning memory behavior. *ACM Trans. Program. Lang. Syst.* 21, 4, 703–746.
- GUPTA, G. AND RAJOPADHYE, S. 2007. The Z-polyhedral model. In *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '07)*. ACM, 237–248.
- HEINE, F. AND SLOWIK, A. 2000. Volume driven data distribution for NUMA-machines. In *Proceedings from the 6th International Euro-Par Conference on Parallel Processing*. 415–424.
- KELLY, W., MASLOV, V., PUGH, W., ROSSER, E., SHPEISMAN, T., AND WONNACOTT, D. 1996. The Omega Library. Tech. rep., Institute for Advanced Computer Studies, University of Maryland, College Park.
- KÖPPE, M., VERDOOLAE, S., AND WOODS, K. 2008. An implementation of the Barvinok-Woods integer projection algorithm. In *Proceedings of the International Conference on Information Theory and Statistical Learning*. 53–59.
- KREISEL, G. AND KREVINE, J. L. 1967. *Elements of Mathematical Logic*. North-Holland Publishing.
- LASARUK, A. AND STURM, T. 2007. Weak quantifier elimination for the full linear theory of the integers: A uniform generalization of Presburger arithmetic. *Appl. Algebra Eng., Comm. Comput.* 18, 6, 545–574.
- LOECHNER, V. 1999. Polylib: A library for manipulating parameterized polyhedra. Tech. rep., LSIIT - ICPS UMR7005 Univ. Louis Pasteur-CNRS.
- LOECHNER, V. AND WILDE, D. K. 1997. Parameterized polyhedra and their vertices. *Int. J. Parallel Program.* 25, 6, 525–549.
- LOECHNER, V., MEISTER, B., AND CLAUS, P. 2002. Precise data locality optimization of nested loops. *J. Supercomput.* 21, 1, 37–76.
- MEISTER, B. 2004. Projecting periodic polyhedra for loop nest analysis. In *Proceedings of the 11th Workshop on Compilers for Parallel Computers (CPC '04)*. 13–24.
- NOOKALA, S. P. K. AND RISSET, T. 2000. A library for Z-polyhedral operations. Tech. rep. 1330, Irisa.
- PARKER, E. AND CHATTERJEE, S. 2004. An automata-theoretic algorithm for counting solutions to Presburger formulas. In *Proceedings of the 13th International Conference on Compiler Construction*. Lecture Notes in Computer Science Series, vol. 2985, 104–119.
- PUGH, W. 1991. The Omega test: A fast and practical integer programming algorithm for dependence analysis. In *Proceedings of the ACM/IEEE Conference on Supercomputing*. ACM Press, 4–13.
- PUGH, W. 1994. Counting solutions to Presburger formulas: how and why. In *Proceedings of the SIGPLAN Conference on Programming Language Design and Implementation (PLDI '94)*. 121–134.
- PUGH, W. AND WONNACOTT, D. 1994. Experiences with constraint-based array dependence analysis. In *Principles and Practice of Constraint Programming*. 312–325.
- QUINTON, P., RAJOPADHYE, S., AND RISSET, T. 1997. On manipulating Z-polyhedra using a canonical representation. *Parallel Process. Lett.* 7, 2, 181–194.
- RAMANUJAM, J., HONG, J., KANDEMIR, M., AND NARAYAN, A. 2001. Reducing memory requirements of nested loops for embedded systems. In *Proceedings of the 38th Conference on Design Automation (DAC '01)*. ACM Press, New York, NY, 359–364.
- SCHRIJVER, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons.
- SEGHIR, R. AND LOECHNER, V. 2006. Memory optimization by counting points in integer transformations of parametric polytopes. In *Proceedings of the International Conference on Compilers, Architectures, and Synthesis for Embedded Systems (CASES '06)*. 74–82.

- TEICH, J. AND THIELE, L. 1993. Partitioning of processor arrays: a piecewise regular approach. *Integr. VLSI J.* 14, 3, 297–332.
- TURJAN, A., KIENHUIS, B., AND DEPRETTERE, E. 2005. Solving out-of-order communication in Kahn process networks. *J. VLSI Signal Process. Syst.* 40, 1, 7–18.
- VERDOOLAEGE, S. 2006. Barvinok: user guide. <http://www.kotnet.org/~skimo/barvinok/>.
- VERDOOLAEGE, S. AND WOODS, K. 2005. Counting with rational generating functions. <http://arxiv.org/abs/math/0504059>.
- VERDOOLAEGE, S., BEYLS, K., BRUYNNOOGHE, M., AND CATTHOOR, F. 2005. Experiences with enumeration of integer projections of parametric polytopes. In *Proceedings of the 14th International Conference on Compiler Construction*, R. Bodik, Ed., Vol. 3443, Springer, 91–105.
- VERDOOLAEGE, S., SEGHIR, R., BEYLS, K., LOECHNER, V., AND BRUYNNOOGHE, M. 2004. Analytical computation of Ehrhart polynomials: Enabling more compiler analyses and optimizations. In *Proceedings of the International Conference on Compilers, Architectures, and Synthesis for Embedded Systems*. 248–258.
- VERDOOLAEGE, S., SEGHIR, R., BEYLS, K., LOECHNER, V., AND BRUYNNOOGHE, M. 2007. Counting integer points in parametric polytopes using Barvinok’s rational functions. *Algorithmica* 48, 1, 37–66.
- ZHAO, Y. AND MALIK, S. 2000. Exact memory size estimation for array computations. *IEEE Trans. VLSI Syst.* 8, 5, 517–521.
- ZHU, H., LUICAN, I. I., AND BALASA, F. 2006. Memory size computation for real-time multimedia applications based on polyhedral decomposition. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.* E89-A, 12, 3378–3386.

Received May 2010; revised May 2011, November 2011; accepted November 2011