The Journal of Supercomputing, 31, 111–135, 2005

A Polynomial-Time Dependence Test for Determining Integer-Valued Solutions in Multi-Dimensional Arrays Under Variable Bounds

WENG-LONG CHANG
Department of Information Management, Southern Taiwan University of Technology, Tainan County, Taiwan 710, Republic of China

CHIH-PING CHU chucp@csie.ncku.edu.tw
JIA-HWA WU
Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan 701, Republic of China

Abstract. Multi-dimensional arrays with linear subscripts occur quite frequently in real programs. For *multi-dimensional linear* arrays under *variable* bounds as well as any given *direction vectors*, the generalized Lambda test is an efficient and precise data dependence method to check whether there exist *real-valued* solutions. In this paper, we propose a multi-dimensional generalized interval test—a polynomial-time dependence test that can be applied towards testing whether there are *integer-valued* solutions for *multi-dimensional linear* arrays under *variable* limits and any given *direction vectors*. Experimental results with benchmark showing the effects of the multi-dimensional generalized interval test over the generalized Lambda test are also presented.

Keywords: parallelizing/vectorizing compiler, data dependence analysis, loop parallelization, loop vectorization

1. Introduction

The question of whether multi-dimensional array references with linear subscripts may be parallelized/vectorized depends upon the resolution of those multi-dimensional array aliases. The resolution of multi-dimensional array aliases is to ascertain whether two references to the same multi-dimensional array within a general loop may refer to the same element of that multi-dimensional array. If the two references to the same multi-dimensional array within a general loop do not refer to the same element, then the general loop can be executed in parallel. Otherwise, it will be executed sequentially. This problem in general case can be reduced to that of checking whether a system of m linear equations with n unknown variables has a simultaneous integer solution, which satisfies the constraints for each variable in the system. It is assumed that m linear equations in a system are written as

$$\begin{aligned} a_{1,1}X_1 + a_{1,2}X_2 + \cdots + a_{1,n-1}X_{n-1} + a_{1,n}X_n &= a_{1,0} \\ &\vdots \\ a_{m,1}X_1 + a_{m,2}X_2 + \cdots + a_{m,n-1}X_{n-1} + a_{m,n}X_n &= a_{m,0}, \end{aligned} \tag{1.1}$$

```
   DO J = 1, 10
     DO I =1, 10
S:  A(I+10*(J-1), I+10*(J-1)) = B (I+10*(J-1), I+10*(J-1))
     ENDDO
   ENDDO
```

Figure 1. A nested do-loop in Fortran language.

where each $a_{i,j}$ is a constant integer for $1 \leq i \leq m$ and $0 \leq j \leq n$. It is postulated that the constraints to each variable in (1.1) are represented as

$$P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \leq X_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s, \tag{1.2}$$

where $P_{r,0}$, $Q_{r,0}$, $P_{r,s}$ and $Q_{r,s}$ are constant integers for $1 \leq r \leq n$. That is, the bounds for each variable X_r are variable.

If each of $P_{r,s}$ and $Q_{r,s}$ is zero in the limits of (1.2), then (1.2) will be reduced to

$$P_{r,0} \leq X_r \leq Q_{r,0}, \quad \text{where } 1 \leq r \leq n. \tag{1.3}$$

That is, the bounds for each variable X_r are constants. Let us use an example to make clear the illustrations stated above. Consider the nested do-loop in Figure 1.

The lower and upper bounds of the first (outer) loop and the second (inner) loop are, respectively, 1 and 10. Therefore, the bounds of the do-loop are constants. This do-loop executes 100 iterations by consecutively assigning the values $1, 2, \ldots, 10$ to J and I and executing the body (the statement ***S***) exactly once in each iteration. The net effect of the do-loop execution is then the ordered execution of the statements:

$$\begin{aligned} A(1, 1) &= B(1, 1) \\ A(2, 2) &= B(2, 2) \\ &\cdots \\ A(100, 100) &= B(100, 100). \end{aligned}$$

To ascertain whether two references to the multi-dimensional array A may refer to the same element of A we have to check if the following two linear equations

$$\begin{aligned} &10 \times X_1 - 10 \times X_2 + X_3 - X_4 = 0 \\ &10 \times X_1 - 10 \times X_2 + X_3 - X_4 = 0 \end{aligned}$$

have a simultaneous integer solution under the constant bounds $1 \leq X_1, X_2 \leq 10$, and $1 \leq X_3, X_4 \leq 10$.

There are several well-known data dependence analysis algorithms applicable for *one-dimensional* arrays under *constant* bounds or *variable* bounds: the GCD test [1–3, 30], Banerjee's method [1–3], the I test and the direction vector I test [17, 20, 23–25], the extended I test and the generalized direction vector I test [5, 7] and the interval reduction test [15]. There are also several well-known data dependence analysis algorithms applicable for *multi-dimensional* arrays under *constant* bounds or *variable* bounds: the generalized GCD test [1–3], the Lambda test [19], the generalized Lambda test [8], the multi-dimensional I test [9], the Power test [31] and the Omega test [26]. There are several well-known data dependence analysis algorithms applicable for arrays with linear subscripts with symbolic coefficients or with non-linear subscripts under *symbolic* bounds: the infinity Banerjee test [3, 22], the Range test [4], the infinity Lambda test [6] and the access range test [13, 21].

In this paper, the extended I test and the generalized direction vector I test [5, 7], the Lambda test and the generalized Lambda test are integrated to check whether *m* linear Eq. (1.1) under *variable* bounds and any given *direction vectors* have *integer-valued* solutions (A dependence testing method determining if there exist *integer-valued* solutions is more precise than that determining if there exist *real-valued* solutions). A theoretical analysis explains that we take advantage of the trapezoidal shape of the convex sets derived from *m* linear equations under *variable* limits and any given *direction vectors* in a data dependence testing. An algorithm called *the multi-dimensional generalized interval test* has been implemented and several measurements have also been performed.

The rest of this paper is proffered as follows. In Section 2, the summary accounts of the extended I test and the direction vector I test, the Lambda test and the generalized Lambda test are presented. In Section 3, the theoretical aspects and the worst-case time complexity of the multi-dimensional generalized interval test are described. Experimental results showing the advantages of the multi-dimensional generalized interval test are given in Section 4. Finally, brief conclusions are drawn in Section 5.

2. Background

The summary accounts of the extended I test and the direction vector I test, the Lambda test and the generalized Lambda test are introduced briefly in this section.

2.1. *The extended I test and the generalized direction vector I test*

A linear equation with the bounds of (1.2) as well as any give direction vectors will be said to be *integer* solvable if the equation has an *integer* solution satisfying the bounds of each variable. Definitions 2.1 and 2.2, respectively, define direction vectors and interval equations [3, 17].

Definition 2.1 A vector of the form $\vec{\theta} = (\theta_1, \ldots, \theta_d)$ is termed as a direction vector. The direction vector $(\theta_1, \ldots, \theta_d)$ is said to be the direction vector from $S_1(\vec{i})$ to $S_2(\vec{j})$ if for

$1 \leq k \leq d$, $i_k \theta_k j_k$, i.e., the relation θ_k is defined by

$$\theta_k = \begin{cases} < & \text{if } i_k < j_k, \\ = & \text{if } i_k = j_k, \\ > & \text{if } i_k > j_k, \\ * & \text{the relation of } i_k \text{ and } j_k \text{ can be ignored, i.e., can be any one of } \{<, =, >\}. \end{cases}$$

Definition 2.2 Let $a_1, \dots, a_{n-1}, a_n, L$ and U be integers. A linear equation

$$a_1X_1 + a_2X_2 + \cdots + a_{n-1}X_{n-1} + a_nX_n = [L, U], \tag{2.1}$$

which is referred to as an interval equation, will be used to denote the set of ordinary equations consisting of:

$$\begin{aligned} a_1X_1 + a_2X_2 + \cdots + a_{n-1}X_{n-1} + a_nX_n &= L \\ a_1X_1 + a_2X_2 + \cdots + a_{n-1}X_{n-1} + a_nX_n &= L + 1 \\ &\vdots \\ a_1X_1 + a_2X_2 + \cdots + a_{n-1}X_{n-1} + a_nX_n &= U. \end{aligned}$$

In the following, Definition 2.3 states the definition of the set of all integer intervals. Definition 2.4 describes the definition of the set of all interval equations. Definition 2.5 introduces the definition of the set for the length of the right-hand side interval on every interval equation in the set of all interval equations.

Definition 2.3 Suppose that the constraints of X_r for $1 \leq r \leq n$ are equal to the bounds of (1.2). Let $[b_0 + \sum_{r=1}^{n} b_r X_r, c_0 + \sum_{r=1}^{n} c_r X_r]$ represent the set of all the integer intervals for every X_r to satisfy the bounds of (1.2), where b_0, c_0, b_r and c_r for $1 \leq r \leq n$ are integers. The set of all the integer intervals, A, is denoted to be equal to

$$\left\{ \left[b_0 + \sum_{r=1}^{n} b_r y_r, c_0 + \sum_{r=1}^{n} c_r y_r \right] \middle| P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} y_s \leq y_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} y_s \quad \text{for } 1 \leq r \leq n \right\}.$$

Definition 2.4 Suppose that the constraints of X_r for $1 \leq r \leq n$ are equal to the bounds of (1.2). Let

$$L = b_0 + \sum_{r=1}^{n} b_r X_r \quad \text{and} \quad U = c_0 + \sum_{r=1}^{n} c_r X_r,$$

where $L \leq U$, and b_0, c_0, b_r and c_r for $1 \leq r \leq n$ are integers. Let $a_1, \ldots, a_{n-1}, a_n$ be integers. The following interval equation

$$a_1 X_1 + a_2 X_2 + \cdots + a_{n-1} X_{n-1} + a_n X_n = [L, U], \tag{2.2}$$

which is referred to as a *variable interval equation*, will be used to denote the set of all the interval equations inferred from every variable X_r under the bounds of (1.2). The set of all the interval equations Γ is denoted to be equal to

$$\left\{ a_1 y_1 + \cdots + a_n y_n = \left[b_0 + \sum_{r=1}^{n} b_r y_r, c_0 + \sum_{r=1}^{n} c_r y_r \right] \right| \\ \left. P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} y_s \leq y_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} y_s \quad \text{for } 1 \leq r \leq n \right\}.$$

Definition 2.5 Suppose that the constraints of X_r for $1 \leq r \leq n$ are equal to the bounds of (1.2). Let K represent the left-hand-side expression $\sum_{r=1}^{n} a_r X_r$, $L = b_0 + \sum_{r=1}^{n} b_r X_r$ and $U = c_0 + \sum_{r=1}^{n} c_r X_r$ in the Eq. (2.2), where $L \leq K \leq U$, and b_0, c_0, a_r, b_r and c_r for $1 \leq r \leq n$ are integers. The set Ω for the length of the right-hand side interval on every interval equation in the Eq. (2.2) is denoted to be equal to

$$\left\{ 1 + (c_0 - b_0) + \sum_{r=1}^{n} (c_r - b_r) y_r) \right| P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} y_s \leq y_r \leq Q_{r,0} \\ \left. + \sum_{s=1}^{r-1} Q_{r,s} y_s \quad \text{for } 1 \leq r \leq n \right\}.$$

A variable interval Eq. (2.2) will be said to be integer solvable if one of the equations in the set, which it defines, is integer solvable. The immediate way to determine this is to test if an integer in between L and U is divisible by the GCD of the coefficients of the left-hand-side terms. If each of b_r and c_r is zero for $1 \leq r \leq n$, then the set of all the interval equations Γ only contains one interval equation. The set Γ will be said to be integer solvable if one of the interval equations in the set Γ is integer solvable. If

$$b_0 + \sum_{r=1}^{n} b_r y_r > c_0 + \sum_{r=1}^{n} c_r y_r$$

in every interval equation in the set Γ, then there are no integer solutions for the set Γ. If

$$\sum_{r=1}^{n} a_r y_r < b_0 + \sum_{r=1}^{n} b_r y_r \quad \text{or} \quad \sum_{r=1}^{n} a_r y_r > c_0 + \sum_{r=1}^{n} c_r y_r$$

in every interval equation in the set Γ, then there are no integer solutions for the set Γ. If the expression of the left-hand side for one of the interval equations in the set Γ is zero items, then the set Γ will be said to be integer solvable if and only if

$$b_0 + \sum_{r=1}^{n} b_r y_r \le 0 \le c_0 + \sum_{r=1}^{n} c_r y_r.$$

It is easy to see that a linear Eq. (1.1) is integer solvable if and only if the only interval equation in the set

$$a_1 X_1 + a_2 X_2 + \cdots + a_{n-1} X_{n-1} + a_n X_n = [a_0, a_0]\}$$

is integer solvable.

In light of the extended I test [5] and the generalized direction vector I test [7], if there is the coefficient a_k for one item in a variable interval Eq. (2.2) with a small enough value to justify the movement of the item to the right, then the item is moved to the right and the value of the right-side interval is changed. Repeat the processing until the number of the item in the left-side in the variable interval Eq. (2.2) is reduced to zero. The two tests generate three possible results when they are used to determine integer solutions of the variable Eq. (2.2). The first generated result of 'yes' means that the variable Eq. (2.2) has integer solutions, and the second generated result of 'no' means that there are no integer solutions for the variable Eq. (2.2). The third generated value of 'maybe', on the other hand, shows that the variable Eq. (2.2) has a solution which satisfies the limits on all the variables which the two tests have managed to move to the right-hand side of the variable Eq. (2.2), and might still have a solution which satisfies the limits on the rest of the variables.

2.2. *The Lambda test and the generalized Lambda test*

Coupled references are groups of reference positions sharing one or more index variables [19, 31]. Geometrically, each linear equation in (1.1) defines a hyperplane π in R^n spaces. The intersection S of m hyperplanes corresponds to the common solutions to all linear equations in (1.1). Obviously, if S is empty then there is no data dependence. Inspecting whether S is empty is trivial in linear algebra [3]. In general, the Banerjee inequalities and the Banerjee Algorithm [5] are, respectively, first applied to test each hyperplane in (1.1) under the bounds of (1.3) or in (1.1) with the bounds of (1.2). If every hyperplane intersects V, then the Lambda test or the generalized Lambda test are employed to simultaneously check every hyper-plane with bounds of (1.3) or (1.2), respectively. The tests form linear combinations of coupled references that eliminate one or more instances of index variables when direction vectors are not considered. While direction vectors are considered, the two methods generate new linear combinations that use a pair of relative index variables.

3. The multi-dimensional generalized interval test

Given the data dependence problem of multi-dimensional arrays with linear subscripts with variable bounds and any given direction vectors, we propose a multi-dimensional generalized interval test (generalized I test and generalized direction vector I test). The multi-dimensional generalized interval I test examines a system of linear equations and deduces whether the system has *integer-valued* solutions. The linear equations have to be first transformed by the multi-dimensional generalized interval test to their corresponding *variable interval* equations. It is straightforward that the linear equations are integer solvable if and only if its corresponding variable interval equations are integer solvable.

Assume that there are m variable interval equations written as

$$\begin{aligned}
&a_{1,1}X_1 + a_{1,2}X_2 + \cdots + a_{1,n-1}X_{n-1} + a_{1,n}X_n \\
&\quad = \left[b_0^{(1)} + \sum_{r=1}^{r=n} b_r^{(1)} X_r,\, c_0^{(1)} + \sum_{r=1}^{r=n} c_r^{(1)} X_r \right] \\
&a_{m,1}X_1 + a_{m,2}X_2 + \cdots + a_{m,n-1}X_{n-1} + a_{m,n}X_n \\
&\quad = \left[b_0^{(m)} + \sum_{r=1}^{r=n} b_r^{(m)} X_r,\, c_0^{(m)} + \sum_{r=1}^{r=n} c_r^{(m)} X_r \right],
\end{aligned} \tag{3.1}$$

where each $a_{i,r}$, $b_0^{(i)}$, $b_r^{(i)}$, $\mathrm{c}_0^{(i)}$ and $c_r^{(i)}$ is a constant integer for $1 \le i \le m$ and $1 \le r \le n$. The constraints to each variable in (3.1) are postulated to be

$$P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s, \tag{3.2}$$

where $P_{r,0}$ and $Q_{r,0}$ are constant integers for $1 \le r \le n$, and X_{2k-1} and X_{2k} satisfy constraints of direction vectors for for $1 \le k \le d$, where d is the number of common loops. Let F_i be the i-th variable interval equation in (3.1). Geometrically, F_i consists of $1 + (c_0^{(i)} - b_0^{(i)}) + \sum_{r=1}^{n} (c_r^{(i)} - b_r^{(i)}) y_r$ linear equations, where

$$P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} y_s \le y_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} y_s \quad \text{for } 1 \le r \le n,$$

in which each linear equation is parallel each other. Hence, F_i contains $1 + (c_0^{(i)} - b_0^{(i)}) + \sum_{r=1}^{n} (c_r^{(i)} - b_r^{(i)}) y_r$ hyperplanes in which each hyperplane is parallel each other. The intersection S of m variable interval equations corresponds to the common solutions to all variable interval equations in (3.1). Obviously, if S is empty then there is no data dependence. The bounds of (3.2) and any given direction vectors define a bounded convex set V in R^n. If any of variable interval equations in (3.1) does not intersect V, then obviously S can not intersect V. However, even if every variable interval equation in (3.1) intersects V, it is still possible that S and V are disjoint. It is assumed that two variable interval equations in (3.1), respectively, intersect V. But the intersection of them is outside of V. If one can find a new variable interval equation which contains S but is disjoint from V, then it immediately follows that S and V do not intersect. The following theorem is an extension of Theorem 1 in [6] and guarantees that if S and V are disjoint, then there must be a variable interval equation which consists of S and is disjoint from V. Furthermore, this variable interval equation is a linear combination of equations in (3.1). On the other hand, if S and V intersect, then no such linear combination exists.

Theorem 3.1 *$S \cap V = \Phi$ if and only if there exists one variable interval equation, β, only consisting of one linear equation, which corresponds to a linear combination of equations in* (3.1) :

$$\left\langle \sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X} \right\rangle = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0} \right],$$

where

$$b_0^{(i)} + \sum_{r=1}^{r=n} b_r^{(i)} X_r \leq a_{i,0} \leq c_0^{(i)} + \sum_{r=1}^{r=n} c_r^{(i)} X_r \quad for 1 \leq i \leq m,$$

such that $\beta \cap V = \Phi \cdot \langle \vec{a}_i, \vec{X} \rangle$ denotes the inner product of $\vec{a}_i = (a_{i,1}, \ldots, a_{i,n})$ and $\vec{X} = (X_1, \ldots, X_n)$.

Proof: ($\Leftarrow$) The variable interval equation, β, contains S and is disjoint from V. So we can immediately derive that S is disjoint from V.

($\Rightarrow$) For the convenience of the proof, (3.1) are rewritten as $A * \vec{Y} = \mathrm{O}$, where

$$A = \begin{pmatrix} -a_{1,0} & a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \vdots & \vdots \\ -a_{m,0} & a_{m,1} & \cdots & a_{m,n} \end{pmatrix}_{(m)*(n+1)}, \quad \vec{Y} = \begin{pmatrix} 1 \\ X_1 \\ \vdots \\ X_n \end{pmatrix}_{(n+1)*1},$$

O is a $m * 1$ zero matrix, and

$$b_0^{(i)} + \sum_{r=1}^{r=n} b_r^{(i)} X_r \leq a_{i,0} \leq c_0^{(i)} + \sum_{r=1}^{r=n} c_r^{(i)} X_r$$

for $1 \leq i \leq m$. We can let

$$S = \{(X_1, \ldots, X_n) : A\vec{Y} = \mathrm{O}\},\ V = \{(X_1, \ldots, X_n) : P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \leq X_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s \quad \text{for } 1 \leq r \leq n \text{ and } X_{2k-1} \text{ and } X_{2k}$$

satisfy constraints of direction vectors for for $1 \leq k \leq d$, where d is the number of common loops}, $S' = \{(1, X_1, \ldots, X_n) : \forall(X_1, \ldots, X_n) \in S\}$, and $V' = \{(1, X_1, \ldots, X_n) : \forall(X_1, \ldots, X_n) \in \text{ V}\}$. Because $S \cap V = \Phi$, we can infer $S' \cap V' = \Phi$.

We let $\alpha = Span(\vec{b}_1, \ldots, \vec{b}_m)$, where $\vec{b}_i = (-a_{i,0}, a_{i,1}, \ldots, a_{i,n})$.$\forall\vec{C} \in \alpha$ and $\vec{D} \in S'$, we can obtain the inner product of $\vec{C}$ and $\vec{D}$ as follows

$$\begin{aligned}\langle\vec{C}, \vec{D}\rangle &= \left\langle\sum_{i=1}^{m} \lambda_i * \vec{b}_i, \vec{D}\right\rangle\\ &= \lambda_1(-a_{1,0} + a_{1,1}X_1 + \cdots + a_{1,n}X_n)\\ &\quad + \cdots + \lambda_m(-a_{m,0} + a_{m,1}X_1 + \cdots + a_{m,n}X_n)\\ &= \lambda_1(0) + \cdots + \lambda_m(0)\\ &= 0.\end{aligned}$$

Therefore, we can at once derive that α is the orthogonal complementary space of S'. For any $\vec{Z}$ in V', consider $\vec{P}_Z$, the projection of $\vec{Z}$ on S'. Since $\|\vec{P}_Z - \vec{Z}\|$ is a continuous function on V' and V' is bounded, there must be exist $\vec{Z}_0$ in V' such that $\|\vec{P}_{Z_0} - \vec{Z}_0\| = \min_{\vec{Z}\in V'} \|\vec{P}_Z - \vec{Z}\|$. This is the minimum distance between S' and V'. Since $\vec{Z}_0 - \vec{P}_{Z_0}$ is orthogonal to S', it must be in α. Hence, the equation $\langle\vec{Z}_0 - \vec{P}_{Z_0}, \vec{D}\rangle = 0$ is a linear combination of equations in (3.1), i.e., $\vec{Z}_0 - \vec{P}_{Z_0} = \lambda_1 * \vec{b}_1 + \cdots + \lambda_m * \vec{b}_m$. The equation $\langle\vec{Z}_0 - \vec{P}_{Z_0}, \vec{D}\rangle = 0$ is actually equal to

$$\left\langle\sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X}\right\rangle = \sum_{i=1}^{m} \lambda_i * a_{i,0}.$$

Therefore, the equation,

$$\left\langle\sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X}\right\rangle = \sum_{i=1}^{m} \lambda_i * a_{i,0},$$

is transformed to one new variable interval equation,

$$\left\langle\sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X}\right\rangle = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right].$$

Let β be the new variable interval equation. Hence, we can immediately conclude that the new variable interval equation, β, which contains S. Since $\langle\vec{Z}_0 - \vec{P}_{Z_0}, \vec{Z}\rangle > 0$ for any $\vec{Z}$ in V' (Refer to [6].), so each element $\vec{X}$ in V satisfies

$$\left\langle\sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X}\right\rangle > \sum_{i=1}^{m} \lambda_i * a_{i,0}.$$

Therefore, we can at once derive $\beta \cap V = \Phi$. □

An array $(\lambda_1, \ldots, \lambda_m)$ in Theorem 3.1 determines one variable interval equation that contains S. There are infinite number of such variable interval equations. The tricky part in the multi-dimensional generalized interval test is to examine as few variable interval equations as necessary to determine whether S and V intersect. We start from the case of m = 2, both for convenience of presentation and for practical importance of two-dimensional arrays [12].

3.1. The case of two dimensional array references

In the case of two dimensional array references, two variable interval equations in (3.1) are

$$F_1 = \left[b_0^{(1)} + \sum_{r=1}^{r=n} b_r^{(1)} X_r, c_0^{(1)} + \sum_{r=1}^{r=n} c_r^{(1)} X_r \right]$$

and

$$F_2 = \left[b_0^{(2)} + \sum_{r=1}^{r=n} b_r^{(2)} X_r, c_0^{(2)} + \sum_{r=1}^{r=n} c_r^{(2)} X_r \right]$$

where $F_i = a_{i,1}X_1 + \cdots + a_{i,n}X_n$ for $1 \le i \le 2$. An arbitrary linear combination of the two variable interval equations can be written as $\lambda_1 F_1 + \lambda_2 F_2 = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$, where

$$b_0^{(1)} + \sum_{r=1}^{r=n} b_r^{(1)} X_r \le a_{1,0} \le \mathrm{c}_0^{(1)} + \sum_{r=1}^{r=n} c_r^{(1)} X_r \quad \text{and}$$
$$b_0^{(2)} + \sum_{r=1}^{r=n} b_r^{(2)} X_r \le a_{2,0} \le \mathrm{c}_0^{(2)} + \sum_{r=1}^{r=n} c_r^{(2)} X_r.$$

The domain of (λ_1, λ_2) is the whole R^2 space. Let $F_{\lambda_1,\lambda_2} = \lambda_1 F_1 + \lambda_2 F_2 = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$, that is $F_{\lambda_1,\lambda_2} = -(\lambda_1 a_{1,0} + \lambda_2 a_{2,0}) + (\lambda_1 a_{1,1} + \lambda_2 a_{2,1})X_1 + \cdots + (\lambda_1 a_{1,n} + \lambda_2 a_{2,n})X_n = 0$. By [19], F_{λ_1,λ_2} is viewed in two ways. With (λ_1, λ_2) fixed, F_{λ_1,λ_2} is a linear function of $(X_1, \ldots, X_n)$ in R^n. With $(X_1, \ldots, X_n)$ fixed, it is a linear function of (λ_1, λ_2) in R^2. Furthermore, the coefficient of each variable in F_{λ_1,λ_2} is a linear function of (λ_1, λ_2) in R^2, i.e., $\Psi^{(i)} = \lambda_1 a_{1,i} + \lambda_2 a_{2,i}$ for $1 \le i \le n$ or $\Phi^{(i)} = \lambda_1(a_{1,2i-1} + a_{1,2i}) + \lambda_2(a_{2,2i-1} + a_{2,2i})$ for $1 \le i \le d$. The equation $\Psi^{(i)} = 0$, $1 \le i \le n$, is called a Ψ line in R^2. The equation $\Phi^{(i)} = 0$, $1 \le i \le d$, is called a Φ line in R^2.

A nonempty set $C \subset R^m$ is a cone if $\varepsilon\vec{\lambda} \in C$ for each $\vec{\lambda} \in C$ and $\varepsilon \ge 0$ [30]. It is obvious that each cone contains the zero vector. Moreover, a cone that includes at least one nonzero vector $\vec{\lambda}$ must consist of the"ray"of $\vec{\lambda}$, namely $\{\varepsilon\vec{\lambda} \mid \varepsilon \ge 0.\}$. Such cones can clearly be viewed as the union of rays. By [19], there are at most n Ψ lines and $n/2\Phi$ lines which together divide R^2 into at most $3n$ regions. Each region contains the zero vector. Any one nonzero element $\vec{\lambda}$ and the zero vector in the region form the ray of $\vec{\lambda}$, namely $\{\varepsilon\vec{\lambda} \mid \varepsilon \ge 0\}$.

Therefore, each region can be viewed as the union of the rays. It is very obvious from the definition of cone that each region is a cone [30].

In the following, Lemmas 3.1 to 3.6 are extended from [5, 7, 8, 17, 19]. Definitions 3.1 to 3.2 are cited from [19] directly.

Lemma 3.1 *Suppose that a bounded convex set V is defined simply by the limits of* (3.2). *(The dependence directions will later be taken account of.) If* $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ *is* $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le 2n)$*—integer solvable for every* (λ_1, λ_2) *in every* Ψ *line, then* $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ *is also* $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le n)$*—integer solvable for every* (λ_1, λ_2) *in* R^2.

Proof:

1. From the extended I test in [5], because $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ for $1 \le r \le n)$—integer solvable for every (λ_1, λ_2) in every Ψ line, there must be at least one element in V such that $F_{\lambda_1,\lambda_2} - (\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}) = 0$.
2. We have that $F_{\lambda_1,\lambda_2} - (\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}) = 0$ for any point (λ_1, λ_2) on every Ψ line according to the assumption of the lemma. It is immediately concluded that $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ for $1 \le r \le n)$—integer solvable for every point (λ_1, λ_2) on the boundaries of each cone.
3. Every point in each cone can be expressed as a linear combination of some points on the boundary of the same cone, as being a well-known fact in the convex theory. Any point (λ_5, λ_6) in a cone is assumed to be capable of being represented as $(\varepsilon\lambda_1 + \tau\lambda_3, \varepsilon\lambda_2 + \tau\lambda_4)$, where (λ_1, λ_2) and (λ_3, λ_4) are points in the boundary of the cone and $\varepsilon \ge 0$ and $\tau \ge 0$. Because

$$
\begin{aligned}
F_{\lambda_5,\lambda_6}(X_1, \ldots, X_n) - (\lambda_5 a_{1,0} + \lambda_6 a_{2,0}) &= F_{\varepsilon\lambda_1+\tau\lambda_3,\varepsilon\lambda_2+\tau\lambda_4}(X_1, \ldots, X_n) \\
&\quad - (\varepsilon\lambda_1 + \tau\lambda_3)a_{1,0} - (\varepsilon\lambda_2 + \tau\lambda_4)a_{2,0} \\
&= \varepsilon * \big(F_{\lambda_1,\lambda_2}(X_1, \ldots, X_n) - (\lambda_1 a_{1,0} + \lambda_2 a_{2,0})\big) + \tau * \big(F_{\lambda_3,\lambda_4}(X_1, \ldots, X_n) \\
&\quad - (\lambda_3 a_{1,0} + \lambda_4 a_{2,0})\big) \\
&= \varepsilon * 0 + \tau * 0 \\
&= 0,
\end{aligned}
$$

we thus secure that $F_{\lambda_5,\lambda_6} = [\lambda_5 * a_{1,0} + \lambda_6 * a_{2,0}, \lambda_5 * a_{1,0} + \lambda_6 * a_{2,0}]$ is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ for $1 \le r \le n)$—integer solvable for any point (λ_5, λ_6) in each cone. Of course it is also true in the whole R^2 space. Therefore, for any point (λ_1, λ_2) in R^2 space, $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ for $1 \le r \le n)$—integer solvable in R^n space.

□

Lemma 3.2 *Suppose that a bounded convex set V is defined by the limits of* (3.2) *and any given direction vectors. If* $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ *is* ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d+1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in every* Φ *line, then* $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ *is also* ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d + 1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in* R^2.

Proof: Similar to Lemma 3.1. □

It is indicated from Lemmas 3.1 to 3.2 that variables in one variable interval equation can be moved to the right if the coefficients of variables have small enough values to justify the movement. If all coefficients for variables in one variable interval equation have no sufficiently small values to justify the movements, then Lemmas 3.1 to 3.2 can not be applied. While every variable in a variable interval equation can not be moved to the right, Lemmas 3.3 to 3.6 describe a transformation using the GCD test which may enable additional variables to be moved.

Lemma 3.3 *Suppose that a bounded convex set V is defined simply by the limits of* (3.2). *Let* $g = \gcd(\lambda_1 a_{1,1} + \lambda_2 a_{2,1}, \ldots, \lambda_1 a_{1,n} + \lambda_2 a_{2,n})$. *If* $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ *is* ($P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in every* Ψ *line, then* $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ *is also* ($P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in* R^2.

Proof: Similar to Lemma 3.1. □

Lemma 3.4 *Suppose that a bounded convex set V is defined by the limits of* (3.2) *and any given direction vectors. Let* $g = \gcd(\lambda_1 a_{1,1} + \lambda_2 a_{2,1}, \ldots, \lambda_1 a_{1,n} + \lambda_2 a_{2,n})$. *If* $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ *is* ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d + 1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in every* Φ *line, then* $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ *is also* ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d + 1 \le r \le n$)*—integer solvable for every* (λ_1, λ_2) *in* R^2.

Proof: Similar to Lemma 3.1. □

Lemma 3.5 *Suppose that a bounded convex set V is denoted simply by the limit of* (3.2). *Let* $g = \gcd(\lambda_1 a_{1,1} + \lambda_2 a_{2,1}, \ldots, \lambda_1 a_{1,n} + \lambda_2 a_{2,n})$. *Given a line in* R^2 *corresponding to an equation* $a\lambda_1 + b\lambda_2 = 0$, *if* $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ *or* $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ *is* ($P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le n$)*—integer solvable* *in R^n space for any fixed point $(\lambda_1^0, \lambda_2^0) \neq (0,0)$ in the line, then for every (λ_1, λ_2) in the line, $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ or $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil]$ is also $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \leq X_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s$ for $1 \leq r \leq n)$—integer solvable in R^n space*

Proof: Similar to Lemma 3.1. □

Lemma 3.6 *Suppose that a bounded convex set V is denoted by the limit of* (3.2) *and any given direction vectors. Let $g = \gcd(\lambda_1 a_{1,1} + \lambda_2 a_{2,1}, \ldots, \lambda_1 a_{1,n} + \lambda_2 a_{2,n})$. Given a line in R^2 corresponding to an equation $a\lambda_1 + b\lambda_2 = 0$, if $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ or $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ is $(P_{2k,0} \leq X_{2k-1}\theta_k X_{2k} \leq Q_{2k,0}$ for $1 \leq k \leq d$, and $P_{r,0} \leq X_r \leq Q_{r,0}$, for $2d+1 \leq r \leq n)$—integer solvable in R^n space for any fixed point $(\lambda_1^0, \lambda_2^0) \neq (0,0)$ in the line, then for every (λ_1, λ_2) in the line, $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$ or $(1/g) * F_{\lambda_1,\lambda_2} = [\lceil(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rceil, \lfloor(\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0})/g\rfloor]$ is also $(P_{2k,0} \leq X_{2k-1}\theta_k X_{2k} \leq Q_{2k,0}$ for $1 \leq k \leq d$, and $P_{r,0} \leq X_r \leq Q_{r,0}$, for $2d+1 \leq r \leq n)$—integer solvable in R^n space*

Proof: Similar to Lemma 3.1. □

Definition 3.1 Given an equation of the form $a\lambda_1 + b\lambda_2 = 0$ where a, b are not zero simultaneously, a canonical solution of the equation is defined as follows:

$$
\begin{aligned}
&(\lambda_1, \lambda_2) = (1, 0), && \text{if } a = 0;\\
&(\lambda_1, \lambda_2) = (0,\ 1), && \text{if } b = 0;\\
&(\lambda_1, \lambda_2) = (b, -a), && \text{if neither of } a, b \text{ is zero.}\\
&(\lambda_1, \lambda_2) = (1, 1), && \text{if both of } a \text{ and } b \text{ are zero.}
\end{aligned}
$$

Definition 3.2 The Λ set is denoted to be the set of all canonical solutions to Ψ equations and Φ equations. Each element, (λ_1, λ_2), in the Λ set corresponds to one variable interval equation $F_{\lambda_1,\lambda_2} = [\lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}, \lambda_1 * a_{1,0} + \lambda_2 * a_{2,0}]$.

By [19], there are at most $n\Psi$ equations and $\frac{n}{2}\Phi$ equations if V is denoted by the bounds of (3.2) and any given direction vectors. Each of Ψ and Φ equations generates a canonical solution according to Definition 3.1. Each canonical solution forms a new variable interval equation, only containing the only linear equation in light of Definition 3.2. Obviously, new variable interval equations tested are at most $\frac{3\times n}{2}$ if V is defined by the constraints of (3.2) and any given direction vectors.

The multi-dimensional generalized interval test is employed to simultaneously check every variable interval equation. It examines the subscripts from two dimensions, and then figures out the Λ set from Ψ and Φ equations. Each element in the Λ set determines a new variable interval equation. The new variable interval equation is tested to see if it intersects V, by moving variables in one variable interval equation as done in the generalized direction vector I test [7] for testing each *single* dimension.

We now use an example to explain how the multi-dimensional generalized interval I test works. Consider the following equations

$$X_1 + X_2 = 10$$
$$X_1 - X_2 = -2$$

subject to the bounds and given direction vectors

$$1 \le X_1 \le 100, \quad 1 \le X_2 \le 100 \quad \text{and} \quad X_1 < X_2.$$

According to in Definition 2.1, the constraints for X_1 and X_2 will be redefined by $1 \le X_1 \le 99$, and $1 + X_1 \le X_2 \le 100$. According to Definition 3.1, the Γ equations have one canonical solutions (0, 1). According to Definition 3.2, one canonical solution (0, 1) yields the following variable interval equations:

$$X_1 - X_2 = [-2, -2] \tag{Ex1}$$

According to Definition 2.4, the set of all the interval equations Ψ for the variable interval equation (Ex1) is equal to

$$\{X_1 - X_2 = [-2, -2] | \text{every variable} X_r \text{ satisfies its bounds for} 1 \le r \le 2\}.$$

The set Ω for the length of the right-hand side interval on every variable interval equation in the set Γ is $\{1\}$. Therefore, the maximum element in the set Ω is one. It is obvious from Definition 2.4 that the set Γ is integer solvable if the only variable interval equation in the set Γ is integer solvable. The coefficient for X_2 satisfies the assumption of the extension of the direction vector I test [7]: (1) $-1 < 0$, (2) $-1 \le 0 \le 0$, (3) $-1 \le 0 \le 0$ and (4) the absolute value of the coefficient is equal to one. The extension of the direction vector I test is applied towards moving the term X_2 to the right-hand side of the only variable interval equation in the set Γ. The new set Γ_1 in light of the extension of the direction vector I test and Definition 2.4 is

$$\{X_1 = [X_1 - 1,\ 98] | \text{every variable } X_r \text{ satisfies its bounds for } 1 \le r \le 1\}.$$

Now the set Ω_1 for the length of the right-hand side interval on every interval equation in the set Γ_1 is equal to $\{100 - X_1 | 1 \le X_1 \le 99\}$. The maximum element computed by the Banerjee algorithm [1] in the set Ω_1 is 99. When the maximum element is 99, the value for X_1 is equal to 1. Because $X_1 = 1$ and $1 \le X_1 \le 99$, so $X_1 - 1 \le X_1 \le 98$ hold. Therefore, there exists a constant interval equation in the set Γ_1 satisfying the given limitations. The coefficient for X_1 satisfies the assumption of the extension of the direction vector I test: (1) $1 > 0$, (2)$1 \ge 1 \ge 0$, (3)$1 \ge 0 \ge 0$, and (4) the value of the coefficient is less than 99.

The extension of the direction vector I test is employed toward moving the term $-X_1$ to the right. The new set Γ_2 is

$$\{0 = [-1, 98 - X_1] \mid 1 \leq X_1 \leq 99\}.$$

The expression of the left-hand side on the variable interval equation in the set Γ_2 is reduced to zero items. The variable interval equation in the set Γ_2 is integer solvable because $-1 \leq 0 \leq 98 - X_1$ hold for $1 \leq X_1 \leq 99$. Therefore, the multi-dimensional generalized interval test in light of Lemmas 3.1 to 3.6 infers that there is *integer*-valued solution.

3.2. *The case of multi-dimensional array references*

We take account of m interval equations in (3.1) with $m > 2$ for generalizing the multi-dimensional generalized interval test. All m variable interval equations are assumed to be connected; otherwise they can be partitioned into smaller systems. As stated before, we can hypothesize that there are no redundant equations. An arbitrary linear combination of m variable interval equations in (3.1) can be written as

$$F_{\lambda_1,\ldots,\lambda_m} = \left\langle \sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X} \right\rangle = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0} \right],$$

where

$$b_0^{(i)} + \sum_{r=1}^{r=n} b_r^{(i)} X_r \leq a_{i,0} \leq \mathrm{c}_0^{(i)} + \sum_{r=1}^{r=n} c_r^{(i)} X_r \quad \text{for } 1 \leq i \leq m \text{ and } \quad \langle \vec{a}_i, \vec{X} \rangle$$

denotes the inner product of $\vec{a}_i = (a_{i,1}, \ldots, a_{i,n})$ and $\vec{X} = (X_1, \ldots, X_n)$.

Assume that

$$g = \gcd\left(\sum_{i=1}^{m} \lambda_i * a_{i,1}, \ldots, \sum_{i=1}^{m} \lambda_i * a_{i,n} \right).$$

It is to be determined whether

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0} \right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0} \right) \middle/ g \right\rceil, \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0} \right) \middle/ g \right\rfloor \right]$$

is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ for $1 \le r \le 2n)$—integer solvable in R^n space for arbitrary $(\lambda_1, \ldots, \lambda_m)$,or it is to be tested if

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rceil, \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rfloor\right]$$

is $(P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ for $1 \le k \le d$, and $P_{r,0} \le X_r \le Q_{r,0}$, for $2d + 1 \le r \le n)$—integer solvable in R^n space for arbitrary $(\lambda_1, \ldots, \lambda_m)$. By [19], the coefficient of each variable in $F_{\lambda_1,\ldots,\lambda_m}$ is a linear function of $(\lambda_1, \ldots, \lambda_m)$ in R^m, which is $\Psi^{(i)} = \sum_{j=1}^{m} \lambda_j a_{j,i}$ for $1 \le i \le n$ and $\Phi^{(i)} = \sum_{j=1}^{m} \lambda_j(a_{j,2i-1} + a_{j,2i})$ for $1 \le i \le d$. The equation $\Psi^{(i)} = 0$, $1 \le i \le n$, is called a Ψ equation (also called λ line). The equation $\Phi^{(i)} = 0, 1 \le i \le d$, is called a Φ equation (also called λ line).

The following Lemmas are extended from [5, 7, 8, 17, 19].

Lemma 3.7 *Suppose that a bounded convex set V is defined simply by the limits of* (3.2). *Let*

$$g = \gcd\left(\sum_{i=1}^{m} \lambda_i * a_{i,1}, \ldots, \sum_{i=1}^{m} \lambda_i * a_{i,n}\right).$$

If

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rceil, \quad \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rfloor\right]$$

is $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s}X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s}X_s$ *for* $1 \le r \le n)$*—integer solvable for every* $(\lambda_1, \ldots, \lambda_m)$ *in every* λ *line, then*

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rceil, \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rfloor\right]$$

is also $(P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \leq X_r \leq Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s$ *for* $1 \leq r \leq n)$*—integer solvable for every* $(\lambda_1, \ldots, \lambda_m)$ *in* R^m *space.*

Proof: Similar to Lemma 3.1. □

Lemma 3.8 *Suppose that a bounded convex set V is defined by the limits of* (3.2) *and any given direction vectors. Let*

$$g = \gcd\left(\sum_{i=1}^{m} \lambda_i * a_{i,1}, \ldots, \sum_{i=1}^{m} \lambda_i * a_{i,n}\right).$$

If

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rceil, \quad \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rfloor\right]$$

is $(P_{2k,0} \leq X_{2k-1}\theta_k X_{2k} \leq Q_{2k,0}$ *for* $1 \leq k \leq d$*, and* $P_{r,0} \leq X_r \leq Q_{r,0}$*, for* $2d + 1 \leq r \leq n)$*—integer solvable for every* $(\lambda_1, \ldots, \lambda_m)$ *in every* λ *line, then*

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rceil, \quad \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \middle/ g \right\rfloor\right]$$

is also $(P_{2k,0} \leq X_{2k-1}\theta_k X_{2k} \leq Q_{2k,0}$ *for* $1 \leq k \leq d$*, and* $P_{r,0} \leq X_r \leq Q_{r,0}$*, for* $2d + 1 \leq r \leq n)$*—integer solvable for every* $(\lambda_1, \ldots, \lambda_m)$ *in* R^m *space.*

Proof: Similar to Lemma 3.1. □

Lemma 3.9 *Suppose that a bounded convex set V is defined simply by the limits of* (3.2). *Given a line in R^m which crosses the origin of the coordinates and let*

$$g = \gcd\left(\sum_{i=1}^{m} \lambda_i * a_{i,1}, \dots, \sum_{i=1}^{m} \lambda_i * a_{i,n}\right).$$

If

$$F_{\lambda_1,\dots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\dots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rceil, \quad \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rfloor\right]$$

is ($P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s$ *for* $1 \le r \le n$)*—integer solvable in* R^n *space for any fixed point* $(\lambda_1^0, \dots, \lambda_m^0) \neq (0, \dots, 0)$ *in the line*, *then for every* $(\lambda_1, \dots, \lambda_m)$ *in the line,*

$$F_{\lambda_1,\dots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\dots,\lambda_m} = \left[\left\lceil \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rceil, \quad \left\lfloor \left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right) \Big/ g \right\rfloor\right]$$

is also ($P_{r,0} + \sum_{s=1}^{r-1} P_{r,s} X_s \le X_r \le Q_{r,0} + \sum_{s=1}^{r-1} Q_{r,s} X_s$ *for* $1 \le r \le n$)*—integer solvable in* R^n *space*

Proof: Similar to Lemma 3.3. □

Lemma 3.10 *Suppose that a bounded convex set V is defined by the limits of* (3.2) *and any given direction vectors. Given a line in R^m which crosses the origin of the coordinates and let*

$$g = \gcd\left(\sum_{i=1}^{m} \lambda_i * a_{i,1}, \dots, \sum_{i=1}^{m} \lambda_i * a_{i,n}\right).$$

If

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil\left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right)\Big/ g\right\rceil, \quad \left\lfloor\left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right)\Big/ g\right\rfloor\right]$$

is ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d+1 \le r \le n$)*—integer solvable in* R^n *space for any fixed point* $(\lambda_1^0, \ldots, \lambda_m^0) \ne (0, \ldots, 0)$ *in the line, then for every* $(\lambda_1, \ldots, \lambda_m)$ *in the line,*

$$F_{\lambda_1,\ldots,\lambda_m} = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0}\right]$$

or

$$(1/g) * F_{\lambda_1,\ldots,\lambda_m} = \left[\left\lceil\left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right)\Big/ g\right\rceil, \quad \left\lfloor\left(\sum_{i=1}^{m} \lambda_i * a_{i,0}\right)\Big/ g\right\rfloor\right]$$

is also ($P_{2k,0} \le X_{2k-1}\theta_k X_{2k} \le Q_{2k,0}$ *for* $1 \le k \le d$, *and* $P_{r,0} \le X_r \le Q_{r,0}$, *for* $2d+1 \le r \le n$)*—integer solvable in* R^n *space.*

Proof: Similar to Lemma 3.3. □

The details of the multi-dimensional generalized interval test in the general case is not considered here since the discussion is similar to the case of $m = 2$.

3.3. The algorithm

For convenience of presentation, the extended I test [5] and the extended direction vector I test [7] are called as the extended interval test. We now summarize the illustration into an algorithm. The algorithm is described below.

Input: *M* interval equations (3.1), the constraints (3.2) to each variable in (3.1) and a set of λ values $(\lambda_1, \ldots, \lambda_m)$.

Output:

no: Equations (3.1) under the constraints (3.2) have no *integer-valued* solutions.
yes: Equations (3.1) under the constraints (3.2) have *integer-valued* solutions.

maybe: The proposed method cannot conclude whether Eqs. (3.1) under the constraints (3.2) have *integer-valued* solutions.

Method:

Step 1. According to $(\lambda_1, \ldots, \lambda_m)$, we can obtain a new interval equation,

$$F_{\lambda_1,\ldots,\lambda_m} = \left\langle \sum_{i=1}^{m} \lambda_i * \vec{a}_i, \vec{X} \right\rangle = \left[\sum_{i=1}^{m} \lambda_i * a_{i,0}, \sum_{i=1}^{m} \lambda_i * a_{i,0} \right],$$

where $\langle \vec{a}_i, \vec{X} \rangle$ denotes the inner product of $\vec{a}_i = (a_{i,1}, \ldots, a_{i,n})$ and $\vec{X} = (X_1, \ldots, X_n)$.
Step 2. The extended interval test is applied to deal with the new interval equation.
Step 3. If the extended interval test finds there exists an integer solution, then a result of **yes** is returned and the processing is terminated. Otherwise, go to Step 4.
Step 4. If the extended interval test determines there exist no integer solutions, then a result of **no** is returned and the processing is terminated. Otherwise, go to Step 5.
Step 5. The extended interval test cannot be applied to determine if there exists an integer solution. A result of **maybe** is returned and the processing is terminated.

If the proposed method returns a result of **yes** or **no**, then that result is accurate; i.e., a returned value of **yes** means that the equations have integer-valued solutions and a returned value of **no** means that the equations have no integer-valued solutions. A returned value of **maybe**, on the other hand, means that the proposed method does not derive whether the equations have *integer-valued* solutions.

3.4. *Time complexity*

The main phases for the multi-dimensional interval test include (1) calculating λ values and (2) examining each variable interval equation. λ values are easily determined according to Ψ equations, Φ equations and Definition 3.1. It is clear that the time complexity to computing a λ value is $O(y)$ from Definition 3.1, where y is a constant. Each λ value corresponds to one variable interval equation. Each variable interval equation is tested to see if it intersects V, by moving variables in left-hand side of one variable interval equation to right-hand side of the variable interval equation as done in the generalized I test and the generalized direction vector I test for one *single* dimension [5, 7]. The worst-case time complexity for them is $O(n^3 + y * n^2 + n^2)$ [5, 7], where n is the number of variables in variable interval equations. Hence, the time complexity of for the multi-dimensional generalized interval test examining one variable interval equation is derived to be $O(n^3 + y * n^2 + n^2 + y)$. The number of variable interval equations checked in the multi-dimensional generalized interval test is at most

$$\prod_{i=1}^{m} (U_i - L_i + 1) * \binom{n}{m-1},$$

where m is the number of *original* coupled references and L_i and U_i are lower and upper bounds in right-hand side of *original* variable interval equations for $1 \le i \le m$, in light of statements in Sections 3.1 and 3.2 and [19]. Therefore, the worst-case time complexity for the multi-dimensional generalized interval test is immediately inferred to be

$$\mathrm{O}\left(\begin{bmatrix} n \\ m-1 \end{bmatrix} * (n^3 + y * n^2 + n^2 + y) * \left(\prod_{i=1}^{m} (U_i - L_i + 1)\right)\right).$$

Two-dimensional arrays with linear subscripts appear quite frequently in real programs [27]. As the lower and upper bounds are initially the same in right-hand side of an initial variable interval equations in coupled references in real programs, therefore, the number of variable interval equations examined in each two-dimensional array tested is at most $\frac{3n}{2}$ according to statements in Section 3.1. If the multi-dimensional generalized interval test is applied to deal with two-dimensional arrays, then their worst-case time complexity is

$$\mathrm{O}\left(\frac{3n}{2} * (n^3 + n^2 * y + n^2 + y)\right).$$

The worst-case time complexity for the Lambda test and the generalized Lambda test dealing with the same array is, respectively,

$$\mathrm{O}\left(\frac{3n}{2} * \left(n + y\right)\right) \quad \text{and} \quad \mathrm{O}\left(\frac{3n}{2} * \left(n^2 + y\right)\right).$$

However, in general, the efficiency of the multi-dimensional generalized interval test is only slightly poorer than that of the generalized Lambda test, the Lambda test, the extended I test and the generalized direction vector I test because the number of variables, *n,* in the variable interval equations tested is generally very small.

4. Experimental results

We tested the multi-dimensional generalized interval test and performed experiments on Personal Computer Intel Pentium IV through the benchmark codes cited from [11, 18, 28]. 515 pairs of array references were observed to have linear subscripts. Meanwhile, it is also noticed that all of the loop lower bounds are constants and all of the loop upper bounds are variable bounds having at least one *symbolic* constant (*unknown* at compile time). Therefore, in order to test the multi-dimensional generalized interval test those *symbolic* constants are assumed to be constants 100. The choice of 100 is arbitrary. In [22] it is reported that, for the Perfect Benchmarks, data dependence testing results (i.e., number of dependences, independences and unanalyzable subscripts existed in the codes) obtained for the original *symbolic* constant bounds are quite close to that for the assumed *constant* bounds. This means that our assumption to the constant, 100, does not change the dependence results (features) existed in the original benchmark codes. The multi-dimensional generalized interval test is only applied to test those arrays with linear subscripts and under variable bounds.

Table 1. Testing capability of the multi-dimensional generalized interval test and the generalized Lambda test for 515 pairs of linear benchmark array references

Testing method	Pairs of arrays tested	The number of *integer-valued* solutions	The number of *non-integer-valued* solutions	Maybe*	Accuracy rate
The proposed method	515	6	36	473	8.1%
The generalized Lambda test	515	0	36	479	7.0%

Maybe*: The testing methods can not generate definitive results for these arrays tested.

The results obtained (Table 1) reveal the multi-dimensional generalized interval test determined that there were 36 integer-valued solutions and 6 non-integer-valued solutions. This implies that there were *definitive* results for 42 pairs of linear arrays. The "accuracy rate" in Table 1 refers to, when given a set of linear subscripts with variable bounds, how often the multi-dimensional generalized interval test detects a case where there is a definitive solution. Let b be the number of the coupled subscripts with variable bounds found in our experiments, and let c be the number that is detected to have definitive solutions. Thus the accuracy rate is denoted to be equal to c/b. In our experiments, 515 pairs of array references checked were found to have linear subscripts with variable bounds, and 42 of them were found to have definitive solutions. So the accuracy rate for the multi-dimensional generalized interval test was about 8.1%.

We also implemented the generalized Lambda test based on [8] to compare their effects with that of the proposed method. The generalized Lambda test was applied to resolve the same 515 pairs of multi-dimensional coupled arrays. It is very clear from the result shown in Table 1 that the generalized Lambda test determined that there were *definitive* solutions for 36 pairs of linear coupled arrays. Let d be the number of the coupled subscripts with variable bounds found in our experiments, and let e be the number that is detected by the proposed method and is not checked by the generalized Lambda test to have definitive solutions. Thus, the improvement rate is denoted to be equal to e/d. In our experiments, 515 pairs of array references were found to have linear subscripts with variable bounds, and 6 of them were found by the proposed method and were not found by the generalized Lambda test to have definitive solutions. So the improvement rate of the multi-dimensional generalized interval test over the generalized Lambda test was about 1.2%. In our experiments, 6 of them are from six different subroutines. This implies that the speedup of the six different programs can be significantly improved by the proposed method.

5. Discussions and conclusions

The study in [26] stated that (1) the cost of scanning array subscripts and loop bounds to build a dependence problem was typically 2 to 4 times of the copying cost (the cost of building a system of dependence equations) for the problem, and (2) the dependence analysis cost for more than half of *simple* arrays tested was typically 2 to 4 times of the copying cost, but the dependence analysis cost for other simple arrays and all of the *regular*, *convex* and *complex* arrays tested was more than 4 times of the copying cost. Based on such results we can figure out that, for simple arrays, the analysis cost of data dependence for parallelizing/vectorizing compilation occupies generally about 29% to 57% of total compiling time. But, for complex arrays the analysis cost of dependence testing takes more than 57% of total compiling time. Therefore, enhancing on dependence testing performance may result in significant improvement on compiling performance of a parallelizing/vectorizing compiler.

For finishing dependence analysis of a pair of arrays tested with different direction vectors, the number of the execution to the proposed method is 3^n, where n is the number of common loops in a general loop. For the other testing methods, the number of the execution for dealing with the same problem is also 3^n. In real programs, the number of arrays checked is very huge. This indicates that compiling time for finishing dependence analysis of a real program is to exponentially increase.

When testing multi-dimensional array references with linear subscripts and *constant* or *variable* bounds, the Lambda test and the generalized Lambda test can determine whether real-valued solutions exist. As we know in dependence analysis a testing strategy concluding the existence of real-valued solutions may sometimes lose the precision and results in false dependency. In this paper we propose the multi-dimensional generalized interval test. The multi-dimensional generalized interval test can ascertain whether *integer-valued* solutions exist for array references with linear subscripts and *variable* bounds and any given direction vectors. Obviously, the significance of the multi-dimensional generalized interval test lies in that it enhances the testing precision, eliminates the possible false dependency and exploits the degree of loop parallelization and vectorization. This is to say that after false dependency are removed in a program from the proposed method a compiler can generate efficient optimized codes for improving the efficiency of the execution for the program.

The Power test is a combination of Fourier-Motzkin variable elimination with an extension of Euclid′s GCD algorithm [31]. The Omega test combines new methods for eliminating equality constraints with an extension of Fourier-Motzkin variable elimination [26]. The two tests currently have the highest precision and the widest applicable range in the field of data dependence analysis for testing arrays with linear subscripts. However, the cost of the two tests is very expensive because the worst-case of Fourier-Motzkin variable elimination is exponential in the number of free variables [26, 31]. Wolfe [31] found that using Fourier-Motzkin variable elimination for dependence testing takes from 22 to 28 times longer than the Banerjee test. Wolfe also indicated that the Lambda test is a very precise and efficient method for testing two-dimensional coupled arrays with constant bounds. Banerjee and Kleanthis [1–3, 20, 25] also indicated that the Omega test is a precise but inefficient method. The Range test [4] and the access range test [13, 21] currently have the highest precision and the widest applicable range for checking nonlinear arrays in the field of data dependence testing.

According to the time complexity analysis, the multi-dimensional generalized interval test performs slightly poorer than that of the generalized Lambda test, the Lambda test, the extended I test and the generalized direction vector I test. Therefore, it is suggested that depending on the application domains and environments, the multi-dimensional generalized interval test can be applied independently or together with other famous methods to analyze data dependence for linear-subscript array references.

References

1. U. Banerjee. *Dependence Analysis for Supercomputing*. Kluwer Academic Publishers, Norwell, MA, 1988.
2. U. Banerjee. *Loop Transformations for Restructuring Compilers:The Foundations*, Kluwer Academic Publishers, 1993.
3. U. Banerjee, *Dependence Analysis*. Kluwer Academic Publishers, Norwell, MA, 1997.
4. W. Blume and R. Eigenmann. Nonlinear and symbolic data dependence testing. *IEEE Transaction on Parallel and Distributed Systems*, 9(12):1180–1194, 1998.
5. W.-L. Chang and C.-P. Chu. The extension of the I test. *Parallel Computing*, 24(14):2101–2127, 1998.
6. W.-L. Chang and C.-P. Chu. The infinity lambda test: A multi-dimensional version of Banerjee's infinity test. *Parallel Computing*, 26(10):1275–1295, 2000.
7. W.-L. Chang and C.-P. Chu. The generalized direction vector I test. *Parallel Computing*, 27(8)1117–1144, 2001.
8. W.-L. Chang, C.-P. Chu, and J. Wu. The generalized lambda test: A multi-dimensional version of banerjee's algorithm. *International Journal of Parallel and Distributed Systems and Networks*, 2(2):69–78, 1999.
9. W.-L. Chang, C.-P. Chu, and J.-H. Wu. A multi-dimensional I-test. *Parallel Computing*, 27:1783–1799, 2001.
10. W.-L. Chang, C.-P. Chu, and J.-H. Wu. A precise dependence analysis for multi-dimensional arrays under specific dependence direction. *Journal of Systems and Software*, 63(2):99–112, 2002.
11. J. Dongarra, M. Furtney, S. Reinhardt, and J. Russell. Parallel Loops—A test suite for parallelizing compilers: Description and example results. *Parallel Computing*, 17:1247–1255, 1991.
12. R. Eigenmann, J. Hoeflinger, and D. Padua. On the automatic parallelization of the perfect benchmarks. *IEEE Transactions on Parallel and Distributed Systems*, 9(1):5–23, 1998.
13. J. P. Hoeflinger. Interprocedural parallelization using memory classification analysis. Ph.D. thesis, University of Illinois at Urbana-Champaign, 1998.
14. J. Hoeflinger and Y. Paek. A comparative analysis of dependnece testing mechanisms. *Language and Compiler for Parallel Computing in* 2000, pp. 289–303.
15. T. C. Huang and C. M. Yang. Data dependence analysis for array references. *The Journal of System and Software*, 52:55–65, 2000.
16. D. E. Knuth. *The Art of Computer Programming*, vol. 2, *Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1981.
17. X. Kong, D. Klappholz, and K. Psarris. The I test. *IEEE Transaction on Parallel and Distributed Systems*, 2(3):342–359, 1991.
18. D. Levine, D. Callahan, and J. Dongarra. A comparative study of automatic vectorizing compilers. *Parallel Computing*, 17:1223–1244, 1991.
19. Z. Li, P.-C. Yew, and C.-Q. Zhu. An efficient data dependence analysis for parallelizing compilers. *IEEE Transaction on Parallel and Distributed Systems*, 1(1):26–34, 1990.
20. D. Niedzielski and K. Psarris. An analytical comparison of the I-test and omega test. *LCPC'99: Twelfth International Workshop on Languages and Compilers for Parallel Computing*.
21. Y. Paek. Compiling for distributed memory multiprocessors based on access region analysis. PhD thesis, University of Illinois at Urbana-Champaign, 1997.
22. P. M. Petersen. Evaluation of programs and parallelizing compilers using dynamic analysis techniques. Ph.D. thesis, University of Illinois at Urbana-Champaign, January 1993.
23. K. Psarris, D. Klappholz, and X. Kong. On the accuracy of the banerjee test. *Journal of Parallel and Distributed Computing*, 12(2):152–158, 1991.
24. K. Psarris, X. Kong, and D. Klappholz. The direction vector *i* test. *IEEE Transaction on Parallel and Distributed Systems*, 4(11):1280–1290, 1993.
25. K. Psarris and K. Kyriakopoulos. Data dependence testing in practice. *Proceedings of the 1999 International Conference on Parallel Architectures and Compilation Techniques*, 1999.

26. W. Pugh. A practical algorithm for exact array dependence analysis. *Communication of the ACM*, 35(8):102–114, 1992.
27. Z. Shen, Z. Li, and P.-C. Yew. An empirical study of Fortran programs for parallelizing compilers. *IEEE Transaction on Parallel and Distributed Systems*, 1(3):356–364, 1992.
28. B. J. Smith et al. *Matrix Eigensystem Routines-Eispack Guide*. Springer, Heidelberg, 1976.
29. R. Triolet, F. Irigoin, and P. Feautrier. Direct parallelization of call statements. *Proc. SIGPLAN Symp. Compiler Construction*, Palo Alto, CA, 1986, pp. 176–185.
30. Vaughan and William Jeffrey. A residuals management model of the iron and steel industry: A linear programming approach. Mich.: Univ. Microfilms International, Ann Arbor, 1986.
31. M. Wolfe and C.W. Tseng. The power test for data dependence. *IEEE Transaction on Parallel and Distributed Systems*, 3(5):591–601, 1992.