ISSN 22773061

# ALGORITHM FOR THE ANALYSIS OF EXACTICTY OF BANERJEE TEST

Monica-Iuliana CIACA
Babeş-Bolyai University, str. T. Mihali no.58-60
monica.ciaca@econ.ubbcluj.ro
Loredana MOCEAN
Babeş-Bolyai University, str. T. Mihali no.58-60
loredana.mocean@econ.ubbcluj.ro
Alexandru VANCEA
Babeş-Bolyai University, str. T. Mihali no.58-60
alexandru.vancea@cs.ubbcluj.ro

## ABSTRACT

In compiler theory, the Banerjee test is a dependence test. The Banerjee test assumes that all loop indices are independent, however in reality, this is often not true. The Bannerjee test is a conservative test. That is, it will not break a dependence that does not exist.

This means that the only thing the test can guarantee is the absence of dependence.

This paper proposes an innovative algorithm which allows precise determination of information about dependences and can act in situation where certain cycling limits are known.

## Keywords

Banerjee, data dependences, imprecise test, solvable

## Academic Discipline And Sub-Disciplines

Mathematics applied in Computer Science

## SUBJECT CLASSIFICATION

37N40

## TYPE (METHOD/APPROACH)

Quasi-Experimental; Literary

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 10, No 3
editor@cirworld.com
www.cirworld.com, member.cirworld.com

## INTRODUCTION

In previous researches we presented Banerjee test for the analysis of data dependencies. Generally, it is an imprecise test, supposing the testing of necessary conditions for the existence of of data dependencies. Utility of such a test is highlighted is the case of un-verifying for conditions, situation that leads to the conclusion of data independence. If we are in the case of conditions verification, testing algorithm decides in conservative way the dependence, even if it is not present.

That's why, some researchers have directed their efforts to formulate sufficient conditions for the existence of data dependencies, aimed at improving the accuracy of these algorithms.

Psarris et al. [Psarris91] demonstrated a sufficient condition for Banerjee test accuracy, which we will present in Theorem 2.2.

Our analysis of accuracy testing algorithms for data dependencies has resulted in the original results that we present in 3.3 and also in [Vancea98].

We have shown that the condition of sufficiency becomes under certain circumstances (defined in Lemma 3.3) also a necessary condition for the accuracy of the test Banerjee (Theorem 3.4). This result allowed the implementation of an improved algorithm (Algorithm 4.1) which has two main advantages:

a). allows precise determination of information about dependences, in some cases that traditional tests Banerjee and CMMDC fail;

b). can act in situations where certain cycling limits are not known (for example analysis 3.4.2 is edifying in this sense);

## 1. Preliminaries. Definitions and notations.

**Definition 1.1.**

(I) Let $a_0$, $a_1$, ... , $a_n$ integers. For each k, $1 \le k \le n$, let $L_k$ şi $U_k$ integers, $L_k \le U_k$. The equation:

$$a_1 I_1 + a_2 I_2 + ... + a_n I_n = a_0$$

they say that is ($L_1$,$U_1$; $L_2$,$U_2$; ...; $L_n$,$U_n$) - solvable (or interval solvable, shortly I-solvable) if exists the integers $i_1$, $i_2$, ... , $i_n$ so that

- $a_1 i_1 + a_2 i_2 + ... + a_n i_n = a_0$ and
- for each k, $1 \le k \le n$, $L_k \le i_k \le U_k$.

(II) For each k, $1 \le k \le m$, let $a_{k0}$, $a_{k1}$, $a_{k2}$, ... , $a_{kn}$ integers. For each k', $1 \le k' \le n$ let $L_{k'}$ and $U_{k'}$ integers, $L_{k'} \le U_{k'}$. The system of equations:

$$\begin{gathered} a_{11} I_1 + a_{12} I_2 + ... + a_{1n} I_n = a_{10} \\ a_{21} I_1 + a_{22} I_2 + ... + a_{21n} I_n = a_{20} \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ a_1 I_1 + a_{m2} I_2 + ... + a_{mn} I_n = a_{m0} \end{gathered}$$

they say that is ($L_1$,$U_1$; $L_2$,$U_2$; ...; $L_n$,$U_n$) - I-solvable if exists integers $i_1$, $i_2$, ..., $i_n$ so that

- $a_{11} i_1 + a_{12} i_2 + ... + a_{1n} i_n = a_{10}$
- $a_{21} i_1 + a_{22} i_2 + ... + a_{2n} i_n = a_{20}$

..............................................

- $a_{m1}i_1 + a_{m2}i_2 + \ldots + a_{mn}i_n = a_{m0}$
- for each k', 1 ≤ k' ≤ n, $L_{k'} \le i_{k'} \le U_{k'}$.

(III) Let $a_1, \ldots, a_n$ , L şi U integers. An equation of the form

$$a_1 I_1 + a_2 I_2 + \ldots + a_n I_n = [L,U]$$

is called interval equation and it has solution if exists $a_0 \in [L,U]$ so that

$a_1 I_1 + a_2 I_2 + \ldots + a_n I_n = a_0$ to be I-rezolvable.

Let be the structure of nested cycles below, where the two references to elements of the m-dimensional A vector potential causes a dependency of data.

```
for j_1 := inf_1 to sup_1 do
        for j_2 := inf_2 to sup_2 do
                .        .        .
                for j_r := inf_r to sup_r do
                        .        .        .
                        B := A[f_1(j_1,...,j_r),f_2(j_1,...,j_r),...,f_m(j_1,...,j_r)] ...
                        .        .        .
                        A[g_1(j_1,...,j_r),g_2(j_1,...,j_r),...,g_m(j_1,...,j_r)] := ...
                        .        .        .
                end for
                .        .
        end for
end for
```

We assume for simplicity that:

- ♦ $\forall$k, 1 ≤ k ≤ r, $\inf_k$ and $\sup_k$ are integers and $\inf_k \le \sup_k$;
- ♦ $\forall$k, 1 ≤ k ≤ r, $f_k$ and $g_k$ are linear functions of the form:

$f_k(j_1,\ldots,j_r) = c_{k1}j_1 + c_{k2}j_2 + \ldots + c_{kr}j_r + c_{k0}$ and respectively (3.1)

$g_k(j_1,\ldots,j_r) = d_{k1}j_1 + d_{k2}j_2 + \ldots + d_{kr}j_r + d_{k0}$

where $c_{ki}, d_{ki} \in \mathbb{Z}$, 0≤ i ≤ r.

Let γ = ($\inf_1,\sup_1$; $\inf_2,\sup_2$; ... ;$\inf_r,\sup_r$; $\inf_1,\sup_1$; $\inf_2,\sup_2$; ... ;$\inf_r,\sup_r$). Between those two references of m-dimensional array A, from above, exists a data dependences if and only if the following system of equations is γ-solvable:

$$f_1(j_1^{'}, j_2^{'},..., j_r^{'}) - g_1(j_1^{''}, j_2^{''},..., j_r^{''}) = 0$$
$$\text{......................................................} \qquad (3.2)$$
$$f_m(j_1^{'}, j_2^{'},..., j_r^{'}) - g_m(j_1^{''}, j_2^{''},..., j_r^{''}) = 0$$

The practical procedure of the testing of dependency suppose conservatively the dependence until at least one of the equations of the system ( 3.2) is not I-solvable, the moment in which we deduce the dependence.

Such a method of approach is called subscript-by-subscript testing [Bane93].

So, let us consider an arbitrary equation chose from (3.2), that, in view of (3.1) we write in the form:

$$c_{k1} j_1 + ... + c_{kr} j_r + c_{k0} - d_{k1} j_1 - ... - d_{kr} j_r - d_{k0} = 0$$

Eliminating possibly terms of coefficient 0 and simplifying notation (meaning factorization based on index values), testing step by step will be to determine if an equation of the form:

$$a_1 I_1 + a_2 I_2 + ... + a_n I_n = a_0 \qquad (3.3)$$

is ($L_1$,$U_1$; $L_2$,$U_2$; ... ,$L_n U_n$)-solvable, where $n \leq 2r$ and $a_k \neq 0$, $1 \leq k \leq n$.

Generalization cmmdc test for interval equations is given without proof (which is immediate) in the following theorem.

**Theorem 1.2.** Fie $a_1$, $a_2$, ..., $a_n$ non-zero integers and let U, L integers. We note

g = **cmmdc**($a_1$, $a_2$, ... , $a_n$)

The interval equation $a_1x_1 + a_2x_2 + ... + a_nx_n = [L,U]$ has solution (is I-solvable) if and only if $\lceil L/g \rceil \leq \lfloor U/g \rfloor$.

In [Bane76] it is shown that in the particular case in which all dependence coefficients have absolute value equation 1, Banerjee test is exactly, solving the problem of integers solutions for the concerned period.

The same conclusion is proved in [Li89] where the coefficient (either ak) of the dependence equation has absolute and unitary solution and

$$\max_{1 \leq i \leq n}(|a_i|) \leq U_k - L_k + 1$$

Based on the analysis of [Psarris91] we prove a sufficient condition of Banerjee test accuracy, which condition is less restrictive than the above.

## 2. A sufficient condition for Banerjee test accuracy.

Based on the positive and negative notations of numbers, upper and lower limits calculated Banerjee test for expression $a_1I_1 + a_2I_2 + ... + a_nI_n$

are respectively

$$l_{\inf} = \sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i)$$
$$l_{\sup} = \sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i) \qquad (3.4)$$

The precise wording of the Banerjee test [Bane97] says that if

$$a_0 \in [l_{\inf}, l_{\sup}]$$

then there exists real numbers $r_1, r_2, \dots, r_n$ so that

- $a_1 r_1 + a_2 r_2 + \dots + a_n r_n = a_0$ and
- $\forall\, i \in \mathbf{N}$, $1 \le i \le n$, $L_i \le r_i \le U_i$.

Sufficiency condition will appeal to the whole values assumed by $a_1 l_1 + \dots + a_n l_n$.

**Lema 2.1.** If $a_1, a_2, \dots, a_n$ are integers, and for every k, $1 \le k \le n$, $L_k$ and $U_k$ are integers fulfill the condition $L_k \le U_k$, then :

$$\sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i) - \sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i) = \sum_{i=1}^{n} |a_i|(U_i - L_i).$$

**Demonstration.** For each of the integers a, L and U we have

$(a^+U - a^-L) - (a^+L - a^-U) = a^+U - a^+L = aU - aL = |a|\,(U - L)$ if $a > 0$ and

$(a+U - a-L) - (a+L - a-U) = -a-L - (-a-U) = aL - aU = -|a|\,L + |a|\,U = |a|\,(U - L)$ if $a < 0$, from where results immediately.the above formula.

**Theorem 2.2.** Let $a_1, a_2, \dots, a_n$ integer numbers, not zero. For each k, $1 \le k \le n$, let $L_k$ and $U_k$ integers with $L_k < U_k$. If exists a permutation π of the set {1, 2, ..., n} so that:

- $|a_{\pi(1)}| = 1$ and
- for each j, $2 \le j \le n$,

$$|a_{\pi(j)}| \le 1 + \sum_{k=1}^{j-1} |a_{\pi(k)}|\,(U_{\pi(k)} - L_{\pi(k)})$$

Then for each integer x from the interval

$$\left[\sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i),\ \sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i)\right]$$

exist integers $x_1, x_2, \dots, x_n$ so that

- $a_1x_1 + a_2x_2 + ... + a_nx_n = x$ and
- for each i, $1 \le i \le n$, $L_i \le x_i \le U_i$.

**Demonstration.** We will demonstrate through induction after n.

Case n = 1. In this case let $a_1 = 1$ and let $a_1 = -1$. If $a_1 = 1$, then

$$\sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i) = a_1^+ L_1 – a_1^- U_1 = aL_1 = L_1$$

and

$$\sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i) = a_1^+ U_1 - a_1^- L_1 = a_1U_1 = U_1$$

so we must demonstrate that for every integer $x \in [L1,U1]$, exit x1 with $L1 \le x1 \le U1$, so that a1x1 = x. But this is obviously true for x1 = x. The case a1 = -1 is demonstrated in similar mode.

Suppose now that the theorem is true for n = q-1 and we consider the case that n = q. In this case we have, from the hypothesis of theorem that exists a permutation π of the numbers {1, 2, ..., q} so that:

- $|a_{\pi(1)}| = 1$ and
- for each j, $2 \le j \le q$,

$$|a_{\pi(j)}| \le 1+\sum_{k=1}^{j-1} |a_{\pi(k)}|(U_{\pi(k)} – L_{\pi(k)})$$

and therefore on the induction assumption, for every integer x from the interval

$$[\sum_{i=1}^{q-1} (a_{\pi(i)}^+ L_{\pi(i)} - a_{\pi(i)}^- U_{\pi(i)}), \sum_{i=1}^{q-1} (a_{\pi(i)}^+ U_{\pi(i)} - a_{\pi(i)}^- L_{\pi(i)})]$$

Exist integers $x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(q-1)}$ so that

- $a_{\pi(1)}x_{\pi(1)} + a_{\pi(2)}x_{\pi(2)} + ... + a_{\pi(q-1)}x_{\pi(q-1)} = x$ and
- for each i, $1 \le i \le q-1$, $L_{\pi(i)} \le x_{\pi(i)} \le U_{\pi(i)}$.

We consider two cases: $a_{\pi(q)} > 0$ and $a_{\pi(q)} < 0$.

Case $a_{\pi(q)} > 0$. Let y an arbitrary integer from interval

$$[\sum_{i=1}^{q} (a_{\pi(i)}^+ L_{\pi(i)} - a_{\pi(i)}^- U_{\pi(i)}), \sum_{i=1}^{q} (a_{\pi(i)}^+ U_{\pi(i)} - a_{\pi(i)}^- L_{\pi(i)})].$$

Because $a_{\pi(q)} > 0$, the interval can be describe as:

$$[(\sum_{i=1}^{q-1} (a^+_{\pi(i)} L_{\pi(i)} - a^-_{\pi(i)} U_{\pi(i)})) + a_{\pi(q)}L_{\pi(q)}, (\sum_{i=1}^{q-1} (a^+_{\pi(i)} U_{\pi(i)} - a^-_{\pi(i)} L_{\pi(i)})) + a_{\pi(q)}U_{\pi(q)}]$$

Let

$$L = \sum_{i=1}^{q-1} (a^+_{\pi(i)} L_{\pi(i)} - a^-_{\pi(i)} U_{\pi(i)})$$

and

$$U = \sum_{i=1}^{q-1} (a^+_{\pi(i)} U_{\pi(i)} - a^-_{\pi(i)} L_{\pi(i)}),$$

and we consider the set of intervals

$$\{ [L + a_{\pi(q)}(L_{\pi(q)} + k), U + a_{\pi(q)}(L_{\pi(q)} + k)] \mid 0 \le k \le U_{\pi(q)} - L_{\pi(q)} \}$$

Because $a_{\pi(q)} > 0$, these intervals are in the next sequence of intervals, ordered ascending after the first element:

$$\begin{aligned}
&[L + a_{\pi(q)}L_{\pi(q)}, U + a_{\pi(q)}L_{\pi(q)}],\\
&[L + a_{\pi(q)}(L_{\pi(q)} + 1), U + a_{\pi(q)}(L_{\pi(q)} + 1)],\\
&[L + a_{\pi(q)}(L_{\pi(q)} + 2), U + a_{\pi(q)}(L_{\pi(q)} + 2)],\\
&\qquad\vdots\\
&[L + a_{\pi(q)}U_{\pi(q)}, U + a_{\pi(q)}U_{\pi(q)}].
\end{aligned}$$

The length of each interval is $U + a_{\pi(q)}(L_{\pi(q)} + k) - (L + a_{\pi(q)}(L_{\pi(q)} + k)) + 1 = U - L + 1$.

We consider two successive intervals, $[L + a_{\pi(q)}(L_{\pi(q)} + k), U + a_{\pi(q)}(L_{\pi(q)} + k)]$ şi $[L + a_{\pi(q)}(L_{\pi(q)} + k + 1), U + a_{\pi(q)}(L_{\pi(q)} + k + 1)]$. Exists a distance between those two, if and only if

$$U + a_{\pi(q)}(L_{\pi(q)} + k) + 1 < L + a_{\pi(q)}(L_{\pi(q)} + k + 1),$$

what comes to $a_{\pi(q)} > U - L + 1$, or, because $a_{\pi(q)} > 0$, $|a_{\pi(q)}| > U - L + 1$, of which falsity results from the assumption that for each $j$, $2 \le j \le q$,

$$|a_{\pi(j)}| \le 1 + \sum_{k=1}^{j-1} |a_{\pi(k)}|(U_{\pi(k)} - L_{\pi(k)})$$

combinated with lemma 3.2.1.

So, we have that

$$\bigcup_{k=0}^{N_{\pi(q)}-M_{\pi(q)}} [L + a_{\pi(q)}(L_{\pi(q)} + k), U + a_{\pi(q)}(L_{\pi(q)} + k)] = [L + a_{\pi(q)}L_{\pi(q)}, U + a_{\pi(q)}U_{\pi(q)}].$$

We observe that for at least k, $0 \le k \le U\pi(q) - L\pi(q)$, the y to which we referred above is in the range

$$[L + a_{\pi(q)}(L_{\pi(q)} + k), U + a_{\pi(q)}(L_{\pi(q)} + k)].$$

For some integer r, $0 \le r \le U-L$, we have in this way

$$y = L + a_{\pi(q)}(L_{\pi(q)} + k) + r = L + r + a_{\pi(q)}(L_{\pi(q)} + k)$$

But, $L \le L + r \le U$ so must exist $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(q-1)}$ so that

- $a_{\pi(1)}x_{\pi(1)} + a_{\pi(2)}x_{\pi(2)} + \ldots + a_{\pi(q-1)}x_{\pi(q-1)} = L+ r$ and
- for each i, $1 \le i \le q-1$, $L_{\pi(i)} \le x_{\pi(i)} \le U_{\pi(i)}$.

We have $0 \le k \le U_{\pi(q)}-L_{\pi(q)}$, so $L_{\pi(q)} \le L_{\pi(q)} + k \le U_{\pi(q)}$, and in this way exist $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(q-1)}, x_{\pi(q)} = L_{\pi(q)} + k$ so that

- $a_{\pi(1)}x_{\pi(1)} + a_{\pi(2)}x_{\pi(2)} + \ldots + a_{\pi(q-1)}x_{\pi(q-1)} + a_{\pi(q)}x_{\pi(q)} = a_{\pi(1)}x_{\pi(1)} + a_{\pi(2)}x_{\pi(2)} + \ldots +$

$a_{\pi(q-1)}x_{\pi(q-1)} + a_{\pi(q)}(L_{\pi(q)} + k) = L + r + a_{\pi(q)}(L_{\pi(q)} + k) = y$ şi

- for each i, $1 \le i \le q$, $L_{\pi(i)} \le x_{\pi(i)} \le U_{\pi(i)}$.

Case $a_{\pi(q)} < 0$. The demonstration is similar with that of the previous case.

The theorem 2.3, whose demonstration follows immediately from the theorem 2.2, shows that hypothesis of Theorem 2.2 is a sufficient condition for the test to be accurate Banerjee, i.e., to determine the limits of cycling full solutions and not just real solutions.

**Theorem 2.3.** Let $a_0$ an integer and let $a_1, a_2, \ldots, a_n$ integers not equal zero. For each k, $1 \le k \le n$, let $L_k$ şi $U_k$ integers so that $L_k < U_k$. If exists a permutation of π of the numbers {1, 2, ..., n} so that:

- $|a_{\pi(1)}| = 1$ and
- for each j, $2 \le j \le n$,
-

$$|a_{\pi(j)}| \le 1 + \sum_{k=1}^{j-1} |a_{\pi(k)}| (U_{\pi(k)} - L_{\pi(k)})$$

then

$$a_0 \in [\sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i), \sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i)]$$

if and only if

$$a_1 I_1 + a_2 I_2 + \ldots + a_n I_n = a_0$$

is ($L_1$, $U_1$; $L_2$, $U_2$; ...; $L_n$, $U_n$) – solvable, i.e., Banerjee test search integers solutions for equation $a_1I_1 + a_2I_2 + ... + a_nI_n = a_0$ between limits of cycling.

The empirical results provided in [Shen90] shows that the number of iterations is relatively high for a cycle instead of dependence equations resulting coefficients are usually low, often even denominations.

These empirical results combined with the results of Theorem 2.3 formally prove that the Banerjee test proves accurate in practice, that it causes the whole of the limits of cycling solutions and not just real solutions.

The formulation of sufficiency theorem suggests that practical application of Banerjee's test has at least exponential complexity (factorial of the number of $a_i$ values) as the worst case would be considered all possible permutations of the values of the coefficients.

Conversely, if the conditions of Theorem 3.2.3 are satisfied by arbitrary permutation of the values of ai , these conditions will be met with more than permutation that has these values sorted in ascending order.

Therefore resulting consequence 2.4, that shows that once the coefficients are sorted, dependence testing can be done in linear time relative to the number of coefficients.

**Theorem 2.4.** Let $a_0$ an integer and let $a_1 \le a_2 \le ... \le a_n$ integers nonzero. For each k, $1 \le k \le n$, let $L_k$ and $U_k$ integeres, so that $L_k < U_k$. If:

- $|a_1| = 1$ and
- For each j, $2 \le j \le n$,

$$|a_j| \le 1+\sum_{k=1}^{j-1} |a_k| (U_k - L_k)$$

then

$$a_0 \in [\sum_{i=1}^{n} (a_i^+ L_i - a_i^- U_i), \sum_{i=1}^{n} (a_i^+ U_i - a_i^- L_i)]$$

if and only if

$$a_1I_1 + a_2I_2 + ... + a_nI_n = a_0$$

is ($L_1$, $U_1$; $L_2$, $U_2$; ...; $L_n$, $U_n$) – solvable, i.e., Banerjee test search integers solutions for ecuation $a_1I_1 + a_2I_2 + ... + a_nI_n = a_0$ between the limits of cycling.

Once the coefficients are ordered, testing can be done in linear time relative to the number of coefficients.

## 3. The increased accuracy Banerjee test.

Because we want to approach only the existence or un-existence of integers solution to the equation of dependence, below by [A, B] we mean the set of integers between A and B inclusive.

**Definition 3.1.** We define the operation of adding of two set of integers S and S' like that:

$$S + S' = \{ s+s' \mid s\in S \text{ şi } s'\in S' \}$$

Notice that if S = [L,U] and S' = { $s_1$, $s_2$, ..., $s_n$ } than we have

$$[L,U] + S' = \bigcup_{i=1}^{n} [L + s_i, U + s_i]$$

The next lemma is obvious.

**Lemma 3.2.** If $M \le x \le N$ then $a^+M - a^-N \le ax \le a^+N - a^-M$. These limits are extreme values of function $f(x) = ax$ in the specified region by $M \le x \le N$.

The result of the following lemma is essential for our purpose of this section. Improve test accuracy Banerjee will be possible just based on restriction imposed in lemma ($|a| \le U - L + 1$).

**Lema 3.3.** Let [L,U] an interval with integer limits. Let a, M and N integer numbers so that $M < N$ and let $S = \{ ax \mid x \in \mathbb{Z}$ şi $M \le x \le N \}$. Then

$$[L,U] + S = [L + a^+M - a^- N, U + a^+N - a^- M]$$

if and only if $|a| \le U - L + 1$.

**Demonstration.** For a = 0 lemma is verified in trivial way. Let a > 0. Then, based on the definition of parts of a number we have:

$$[L + a^+M - a^- N, U + a^+N - a^- M] = [L + aM, U + aN]$$

The general form for an element from S is aM + ka, with k = 0, N - M.

Based on definition 3.1 we have

$$[L,U] + S = \bigcup_{k=0}^{N-M} [L + aM + ka, U + aM + ka] \qquad (3.5)$$

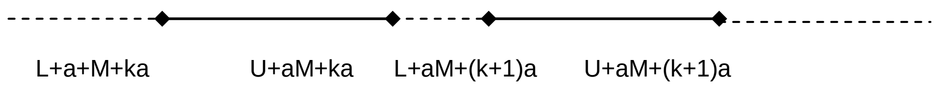
So we must show that:

$$[L,U] + S = [L + aM, U + aN] \quad \Leftrightarrow \quad a \le U - L + 1.$$

To note that $L + aM = L + aM + ka$, for $k = 0$

$U + aN = L + aM + ka$, for $k = N - M$

which means we have to determine a necessary and sufficient condition for the reunion (3.5) to be an interval. This happens, if and only if every two successive intervals are disjoint, or, at worst, are disjoint but adjacent as like the situation:



what is the necessary and sufficient condition

$$L + aM + (k+1)a \le U + aM + ka + 1 \quad \Leftrightarrow \quad a \le U - L + 1$$

Notice that this is true only in the sense that I gave a notation [A, B] at the beginning of the paragraph. If we consider real numbers, of these integers, then the result above does not occur, because the worst "get lost" real values of integers representing the ends of intervals.

The case $a < 0$ is treated in the same way and we obtained $-a \le U - L + 1$. Combining the results we obtained that

$$[L,U] + S = [L + a^+M - a^- N, U + a^+N - a^- M]$$

if and only if $|a| \le U - L + 1$, q.e.d.

Using the result of lemma 3.3, we will show that the condition of sufficiency of theorem 3.3. is also a necessary condition for the accuracy Banerjee test.

**Theorem 3.4.** Let non-zero integers, $a_1, a_2, \dots, a_n$ and $\forall\ k\in \mathbf{N}$, $1 \le k \le n$, f $L_k$ and $U_k$ integers so that $L_k < U_k$. If for each integer x from the interval

$$[\sum_{i=1}^{n}(a_i^+L_i - a_i^-U_i), \sum_{i=1}^{n}(a_i^+U_i - a_i^-L_i)]$$

exist integers $x_1, x_2, \dots, x_n$ so that

- $a_1x_1 + a_2x_2 + \dots + a_nx_n = x$ and
- $\forall\ i\in \mathbf{N}$, $1 \le i \le n$, $L_i \le x_i \le U_i$.

Then exist a permutation π of the set {1, 2, ... , n} so that

(i) $|a_{\pi(1)}| = 1$ and

(ii) $\forall\ j\in\mathbf{N}$, $2 \le j \le n$ $\quad |a_{\pi(j)}| \le 1 + \sum_{k=1}^{j-1} |a_{\pi(k)}|(U_{\pi(k)} - L_{\pi(k)})$

**Demonstration.** We will demonstrate by induction on n. Let n = 1. On basis of hypothesis of theorem, we have that for each x from interval

$$[a_1^+L_1 - a_1^-U_1, a_1^+U_1 - a_1^-L_1]$$

exists an integer $x_1$, $L_1 \le x_1 \le U_1$, so that $a_1x_1 = x$, and from here results that

$$[a_1^+L_1 - a_1^-U_1, a_1^+U_1 - a_1^-L_1] \subset \bigcup_{L_1 \le x_1 \le U_1} [a_1x_1, a_1x_1]$$

On the other hand, according to lemma 3.2 we have that

$$\bigcup_{L_1 \le x_1 \le U_1} [a_1x_1, a_1x_1] \subset [a_1^+L_1 - a_1^-U_1, a_1^+U_1 - a_1^-L_1]$$

so that we have equality:

$$\bigcup_{L_1 \le x_1 \le U_1} [a_1x_1, a_1x_1] = [a_1^+L_1 - a_1^-U_1, a_1^+U_1 - a_1^-L_1]$$

where on the basis of lemma 3.3 in which L = U = 0 we have that |a1| = 1, so for n = 1 the conclusion is verified. We suppose now that the theorem is true for the case n-1 and we deduce the conclusion for the case of n.

Let

$$\begin{aligned} L &= \sum_{i=1}^{n-1} (a_i^+L_i - a_i^-U_i) \\ U &= \sum_{i=1}^{n-1} (a_i^+U_i - a_i^-L_i) \end{aligned} \qquad (3.6)$$

From the hypothesis of induction results that exists a permutation π' of the set {1, 2, ..., n-1} so that

$|a_{\pi'(1)}| = 1$ and

$\forall$ j$\in$**N**, 2 ≤ j ≤ n-1 $|a_{\pi'(j)}| \le 1 + \sum_{k=1}^{j-1} |a_{\pi'(k)}| (U_{\pi'(k)} - L_{\pi'(k)})$

We define a permutation π of the set {1, 2, ... , n} so that:

$$\pi(i) = \begin{cases} \pi'(i), & \text{if} \quad 1 \le i \le n-1 \\ n, & \text{if} \quad i = n \end{cases}$$

And we must show that (for the rest of the value the relation is true through assumption of induction)

$$|a_{\pi(n)}| \le 1 + \sum_{k=1}^{n-1} |a_{\pi(k)}| (U_{\pi(k)} - L_{\pi(k)})$$

but as π(n) = n, it returns to show that:

$$|a_n| \le 1 + \sum_{i=1}^{n-1} |a_i| (U_i - L_i) \qquad (3.7)$$

The hypothesis of the theorem says that for any x in the interval

$$[L+a_n^+L_n-a_n^-U_n, U+a_n^+U_n-a_n^-L_n]$$

exist integers $x_1, x_2, \ldots, x_n$ so that

$$a_1x_1+a_2x_2+\ldots+a_nx_n=x \quad \text{and}$$

$\forall$ i$\in$ **N**, $1 \le i \le n$, $L_i \le x_i \le U_i$.

This assumption combined with lemma 3.3 and Banerjee formulas (3.4,3.6) of calculus of limits for an amount shows that for any x from the interval

$$[L+a_n^+L_n-a_n^-U_n, U+a_n^+U_n-a_n^-L_n]$$

exist integers $w=\sum_{i=1}^{n-1}a_ix_i$ şi $x_n$ , so that

$x = w + a_nx_n$ ;

$L \le w \le U$;

$L_n \le x_n \le U_n$ ;

Hence we have the relation:

$$[L+a_n^+L_n-a_n^-U_n, U+a_n^+U_n-a_n^-L_n] \subset \bigcup_{L_n\le x_n\le U_n}[L+a_nx_n, U+a_nx_n]$$

On the basis of lemma 3.3 we deduce immediately that we have and

$$\bigcup_{L_n\le x_n\le U_n}[L+a_nx_n, U+a_nx_n] \subset [L+a_n^+L_n-a_n^-U_n, U+a_n^+U_n-a_n^-L_n]$$

so that results the equality

$$\bigcup_{L_n\le x_n\le U_n}[L+a_nx_n, U+a_nx_n] = [L+a_n^+L_n-a_n^-U_n, U+a_n^+U_n-a_n^-L_n]$$

And so we can apply the lemma 3.3, from where results

$$|a_n| \le U - L + 1$$

what comes to

$$|a_n| \leq 1 + \sum_{i=1}^{n-1} |a_i| (U_i - L_i)$$

and so the theorem is demonstrated.

## 4. The implementation of algorithm

After the analysis done in this section, the algorithm which reflects the presented results is the following:

**Algorithm 4.1.**

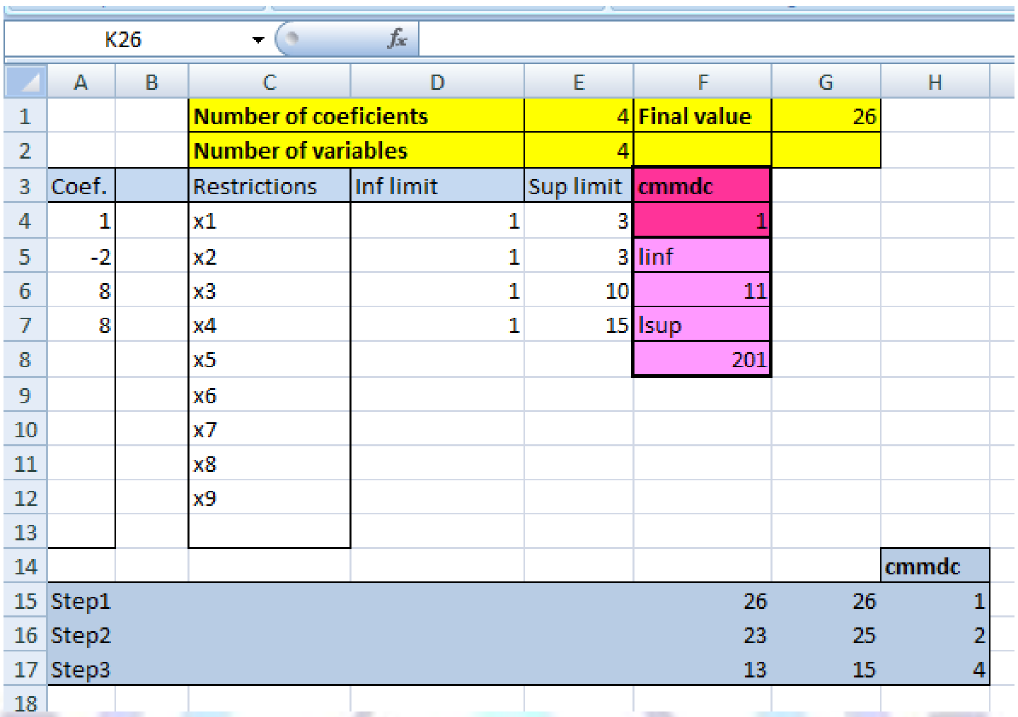
```
input:    a0, a1, ... , an            - equation coefficients of depending;
          L1, U1, ... , Ln, Un   - loop limits;

output:    - NO - dependence equation it is not I-solvable;
           - YES – dependence equation is I-solvable;
           - MAYBE – dependence equation might be I-solvable;

begin
L = a0 ; U = a0 ; coef = { a1, a2, ... , an};
while (true) do
{         while (∃ ai ∈ coef a.î. |ai| ≤ U - L + 1) do
                {
                    $L = L - a_i^+ U_i + a_i^- L_i$ ;
                    $U = U - a_i^+ L_i + a_i^- U_i$ ;
                    coef = coef -{ai};
                    if coef = Ø  then {if (L ≤ 0 and 0 ≤ U) then return ('YES')
                                                            else return ('NO')}
                }
          g = gcd(ai), ai ∈ coef
          if (not(⌈L/g⌉ ≤ ⌊U/g⌋)) then return ('NO');
          if (g ≠ 1) then {for ai ∈ coef  do  ai = ai/g ;
                                L = ⌈L/g⌉; U = ⌊U/g⌋;}
             else return('MAYBE');
}
end.
```

In the next figure we see an implementation of algorithm 4.1. The values are introduced in cells and in final we see the interpretation of results. Our example is explained after the figure.

ISSN 22773061



K26

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | Number of coeficients | | 4 | Final value | 26 | |
| 2 | | | Number of variables | | 4 | | | |
| 3 | Coef. | | Restrictions | Inf limit | Sup limit | cmmdc | | |
| 4 | 1 | | x1 | 1 | 3 | 1 | | |
| 5 | -2 | | x2 | 1 | 3 | linf | | |
| 6 | 8 | | x3 | 1 | 10 | 11 | | |
| 7 | 8 | | x4 | 1 | 15 | lsup | | |
| 8 | | | x5 | | | 201 | | |
| 9 | | | x6 | | | | | |
| 10 | | | x7 | | | | | |
| 11 | | | x8 | | | | | |
| 12 | | | x9 | | | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | cmmdc |
| 15 | Step1 | | | | | 26 | 26 | 1 |
| 16 | Step2 | | | | | 23 | 25 | 2 |
| 17 | Step3 | | | | | 13 | 15 | 4 |
| 18 | | | | | | | | |

**Fig 1: A capture of application**

The value on which we have tested the application are described below. Let dependence equation

$$x_1 - 2x_2 + 8x_3 + 8x_4 = 26$$

with restriction

$$1 \le x_1 \le 3 \qquad 1 \le x_3 \le 10$$
$$1 \le x_2 \le 3 \qquad 1 \le x_4 \le 15 \qquad (3.8)$$

cmmdc(1, -2, 8, 8) = 1, which divide 26 so cmmdc test doesn't exclude the possibility of dependence.

Limit values for expresion $x_1 - 2x_2 + 8x_3 + 8x_4$ compared at conditions (3.8) are calculated according to the formulas of Banerjee (3.4) and we obtain $l_{inf} = 11$ and $l_{sup} = 201$. Because $11 \le 26 \le 201$, Banerjee test also indicate that the equation above can be I-solvable.

After applying the dependence algorithm 4.1. we first obtain:

$$x_1 - 2x_2 + 8x_3 + 8x_4 = [26, 26]$$

and after, because $a_1 = 1 \le U - L + 1 = 26 - 26 + 1 = 1$, we obtain

$$-2x_2 + 8x_3 + 8x_4 = [26-3, 26-1] = [23, 25]$$

cmmdc(-2, 8, 8) = 2 and $12 = \lceil 23/2 \rceil \le \lfloor 25/2 \rfloor = 12$, we continue through rewriting the equation under the form $-x_2 + 4x_3 + 4x_4 = [12, 12]$ and because $|a_2| = 1 \le 12 - 12 + 1 = 1$ we will eliminate the term $x_2$ and we will have:

$$4x_3 + 4x_4 = [12 + 1, 12 + 3] = [13, 15]$$

cmmdc(4, 4) = 4, but this time we have $4 = \lceil 13/4 \rceil \not\leq \lfloor 15/4 \rfloor = 3$, so equation doesn't have solution, so we have dependences.

**So, algorithm 4.1. gives here an exactly answer ( NO dependences ) while CMMDC and Banerjee tests can only assume the conservative (and wrong this time!) dependence.**

It is important to note that in contrast to classical CMMDC and Banerjee tests, test 4.1 may also action in circumstances where certain limits are not known. For example here, if you would not know the variables x3 and x4 limits, the test would conclude dependence gcd (acting solely on the basis of coefficients) and classical Banerjee test is not applicable at all, because it could calculate the limit values. Instead, the application of algorithm 4.1 to 4.2 to above example shows that it is not necessary to know the limits of cycling for x3 and x4, which are not used in the algorithm.

## 5. State of Art

Peterson et al. in their paper [4] include the generalized greatest common divisor test, three variants of Banerjee's test, and the Omega test. Their effectiveness was measured with respect to the Perfect Benchmarks and the linear algebra libraries, EISPACK and LAPACK. Two methods were applied, one using only compile-time information for the analysis, and the second using information gathered during program execution. The results indicate that Banerjee's test is for all practical purposes as accurate as the more complex Omega test in detecting parallelism.

In paper [5], the author proposes a theorem on which delinearization algorithm is based on algorithm itself.

Banerjee et al. in paper [6] present an overview of an automatic programs parallelization and the last section of the paper surveys several experimental studies on the experimental studies on the effectiveness of parallelizing compilers.

In paper [7], the authors discuss the effectiveness of several dependence tests in the Perfect Benchmarks. The tests analyzed include the generalized greatest common divisor

test, Banerjee's test and the Omega test. The dynamic analysis shows that the Omega test does not improve the detected inherent parallelism.

In paper [8], the authors describe the range test that can handle non-linear expressions. The test proves independence by determining whether certain symbolic inequalities hold of a permutation of the loop nest.

## 6. Conclusions

Data dependency analysis is the theoretical basis of the methodology of restructuring sequential programs for automatic parallelization.

Analysis of data dependencies is a problem in the general case un-decidable, Even when we limit the nested loop structures of particular forms (affine structures), determining of a complete information relative to data dependence, even if become decidable, remains an NP-complete problem.

Therefore, in practice are chosen simply techniques, based on the theory of diofantic equations and theory of limits continuous real functions. Even if these methods are imprecise, they generally prove more effective than conventional methods of linear integer programming.

In this context, efforts to improve the accuracy of these algorithms (even under a certain restriction of generality) are justified.

In Section 2 we presented a sufficient condition for the accuracy test demonstrated by Psarris Banerjee et al. in [Psarris91]. In Section 3, which constitutes an original contribution, we have shown that under certain circumstances (as defined in Lemma 3.3) the condition of sufficiency becomes necessary, result which allowed the development of an improved algorithm for data dependence analysis.

Example presented is illustrative regarding the advantages exhibited by this algorithm: the source code as an example to successfully determine the exact dependency information, gcd and Banerjee tests failing in this case. Furthermore, our algorithm can act in situations where the loop limits are not known, our example is illustrative in this aspect.

On the other hand, in the general case, the algorithm 4.1 remains imprecise; there are situations in which he must assume the conservative dependence (variant 'MAYBE' in the algorithm). As we highlight, only a full linear programming algorithm can provide an exact answer for any event.

Improved algorithms for testing the accuracy of data dependencies have been exceptionally important. They highlight the main task of a high degree of parallelism as processed by automatic parallelization algorithms. An analysis of the latter we will do in a further research.

## REFERENCES

[1] Psarris, K., Klappholz, D. and Kong, X.1991. On the accuracy of the Banerjee test, in Journal of Parallel and Distributed Computing, 12, pp.152-157.

[2] Vancea, A. and Boian, F. 1998. On the exactness of a data dependence analysis method, in Studia Informatica, vol.XLIII, no.1, pp.33 - 39.

[3] Shen, Z., Z.Li and Pen-Chung Yew. 1990. An Empirical Study of FORTRAN programs for Parallelizing Compilers, in IEEE Transactions on Parallel and Distributed Systems, vol.1, nr.3, pp.356-364.

[4] Petersen, P.M. and Padua, D.A. 1996. Static and Dynamic Evaluation

of Data Dependence Analysis Techniques IEEE Transactions on parallel and distributed systems, vol. 7, no. 11

[5] Maslov, V. Delinearisation, An Efficient way to break Multiloop Dependence Equations

[6] Banerjee, U., EigenMann, R., Nicolau, A., Padua, D. 1993. Automatic Program Parallelisation

[7] Peterson, P. and Padua, D. Static and Dynamic Evaluations of Data Dependence Analysis.

[8] Blume, W. and Eigenmann, R. The range test, a dependence test for symbolic, non-linear expressions

[9] Banerjee, A., Carrion-i-Silvestre, J.L. Cointegration in panel data with breaks and cross-section dependence

## Author' biography with Photo

**Dr. Monica Iuliana Ciaca** obtained her bachelor's degree at Babes Bolyai University Cluj-Napoca, in the field of Computer Science. After graduation, she has worked as programmer at the Institute for Computation Techniques from Cluj-Napoca.

In 1994 she started working at the Babes Bolyai University as teaching assistant, being interested in artificial intelligence, expert systems, business information systems and software engineering.

She published various articles, the most important being the one written after her participation in a Tempus Phare project, in Perugia. In 2003 she got her PhD in Mathematics and Computer Science, with a thesis on parallel computing: "Implementation Techniques in Parallel Computing".

In the last five years she looked to extend her knowledge in another field: theology. She obtained her Bachelor's Degree and Master's Degree in Biblical Studies and Iconographic exegesis, in 2012, at Babes Bolyai University.

Since 2004 she is Associate Professor at Babes Bolyai University, Cluj-Napoca, Faculty of Economics, in the Department of Business Information Systems.

**Dr. Loredana MOCEAN** has graduated Babes-Bolyai University of Cluj-Napoca, the Faculty of Computer Science, she holds a PhD diploma in Economics and she had gone through didactic position of assistant, lecturer and associate professor, since 2000 when she joined the staff of the Babes- Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration. Also, she graduated Faculty of Economics and Business Administration. She is the author of more than 20 books and over 35 journal articles in the field of Databases, Data mining, Web Ser-vices, Web Ontology, ERP Systems and much more. She is director or member in more than 20 grants and research projects, national and international.

**Dr. Alexandru Vancea** has graduated the Computer Science Department of "Babes-Bolyai" University Cluj-Napoca in 1986 and he obtained Ph.D. in Computer Science in 2000.

Research areas and domains of interests: Programming Languages Design and Analysis, Automatic parallelization of programs, Distributed Programming

Teaching: Operating Systems, Computer Architecture, Fundamentals of Programming Languages.