Affine-by-statement scheduling: extensions for affine dependences and several parameters

Alain Darte*

Laboratoire LIP-IMAG, Ecole Normale Supérieure de Lyon
69364 LYON Cedex 07
e-mail: darte@lip.ens-lyon.fr

June 2, 1992

1 Introduction

This short report is a companion paper of the research report [1]. It cannot be read without it. We present here two extensions of affine-by-statement scheduling for loop nests. The first extension is for loop nests with affine dependences. The second extension is for parametric domains with several size parameters.

2 General problem with affine dependences

The technique presented in [1] for uniform loop nests can be extended to more general cases, namely non perfect loop nests in which dependences are affine. In such a case, different statements can have computation domains of different dimensions. We assume that each point p of the domain is linked to the point $I_{j,i}(p)$ where $I_{j,i}(p) = D_{j,i}p - d_{j,i}$ is an affine function of the

*Supported by the Project C3 of the French Council for Research CNRS, and by the ESPRIT Basic Research Action 3280 "NANA" of the European Economic Community.

indices. The dependence constraints become:

$$\begin{cases} \forall p, A_i p \leq b_i \\ (X_i - X_j D_{j,i})p + X_j d_{j,i} + c_i - c_j \geq 1 \end{cases}$$

which can be replaced by:

$$\begin{cases} X_j D_{j,i} - X_i = X_{j,i} A_i \\ X_{j,i} \geq 0 \\ -X_{j,i} b_i + X_j d_{j,i} + c_i - c_j \geq 1 \end{cases}$$

Example: consider the following program of a matrix-vector product:

```
for i=0 to n
{
    y[i]=0; /* S1 */
    for j=0 to n
    {
        y[i]=y[i]+A[i,j]*x[j]; /* S2 */
    }
}
```

There are two statements:

- S_1 with domain $0 \leq i \leq n$.
- S_2 with domain $0 \leq i, j \leq n$.

The dependences are:

- (i, j) depends on $(i, j-1)$ for S_2 if $j \geq 1$, thus ${}^t(0, 1)$ is a uniform dependence vector on the domain $(0 \leq i \leq n)$ and $(1 \leq j \leq n)$.
- (i, j) of S_2 depends on i of S_1 if $j = 0$, thus $D_{1,2} = (1, 0)$ is an affine dependence on the domain $(0 \leq i \leq n)$ and $j = 0$, i.e:

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} \leq \begin{pmatrix} n \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Thus, for loop nests with affine dependences, a similar theorem as theorem 1 in [1] can be stated. For the example above, the general problem is the following:

$$
\left\{
\begin{array}{l}
X_1 = X_1' \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad X_1 = -X_1'' \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\
X_2 = X_2' \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \quad X_2 = -X_2'' \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \\
X_1' \geq 0 \quad X_1'' \geq 0 \quad X_2' \geq 0 \quad X_2'' \geq 0 \\
\\
X_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \geq 1 \\
\\
X_1(1,0) - X_2 = X_{1,2} \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \\
X_{1,2} \geq 0 \\
-X_{1,2} \begin{pmatrix} n \\ 0 \\ 0 \\ 0 \end{pmatrix} + c_2 - c_1 \geq 1 \\
\\
\gamma_1 - X_1' \begin{pmatrix} n \\ 0 \end{pmatrix} - c_1 \geq 0 \quad \gamma_2 - X_1'' \begin{pmatrix} n \\ 0 \end{pmatrix} + c_1 \geq 0 \\
\gamma_1 - X_2' \begin{pmatrix} n \\ 0 \\ n \\ 0 \end{pmatrix} - c_2 \geq 0 \quad \gamma_2 - X_2'' \begin{pmatrix} n \\ 0 \\ n \\ 0 \end{pmatrix} + c_2 \geq 0 \\
\\
\min(\gamma_1 + \gamma_2)
\end{array}
\right.
$$

The solution found is:

$$
\begin{array}{ll}
X_1 = 0 & c_1 = 0 \\
X_2 = (0,1) & c_2 = 1
\end{array}
$$

which means that the program can be rewritten in a form with explicit parallelism:

```
doall i=0 to n
{
    y[i]=0; /* S1 */
}

for j=0 to n
{
    doall i=0 to n
    {
        y[i]=y[i]+A[i,j]*x[j]; /* S2 */
    }
}
```

Note that for programs with affine dependences, it is no longer true that there exists a linear schedule with a linear execution time. Consider for example, the following program:

```
for i=0 to n
{
    for j=0 to n
    {
        a[i,j]=a[i,j-1]+a[i-1,2j]; /* S1 */
    }
}
```

The limit problem can be unfortunately empty. The set of dependences, considered for $n = \infty$, can not be strictly in one side of an hyperplane, even if it was true for a fixed n (cf figure 1).

3 Limit problem with several parameters

We have seen that, in the case of uniform loops with one parameter N, it was possible to consider only schedules depending linearly on N. In the case of many independent parameters, this technique still holds, even for affine dependences but with some assumptions. In the following, we will consider more general loops with parameters $n_1, \ldots, n_s$, but with the following restrictions:

1. the parameters are independent and nonnegative.

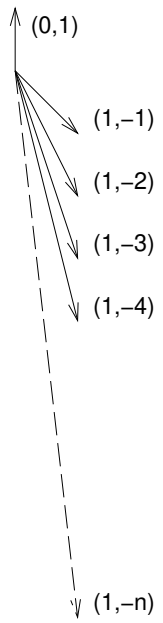

Figure 1: Dependence vectors for the example of an empty limit problem

2. the computation domains are domains $\{Ax \leq b\}$ with $b = n_1 b_1 + \cdots + n_s b_s + b^*$ and $\forall i,\ 1 \leq i \leq s,\ b_i \geq 0$.

3. the domains $\{Ax \leq b\}$ depend on some parameter in each dimension.

4. the dependences are affine and do not depend upon the parameters.

We will consider schedules of the form:

$$\text{the point } p \text{ of statement } i \text{ is scheduled at time:}$$
$$X_i p + n_1 c_{i,1} + \cdots + n_s c_{i,s} + c_i^*$$
$$\text{where } X_i, c_{i,1}, \ldots, c_{i,s}, c_i^* \text{ are constant}$$

Our main result is that if there exists such a solution, then this class of schedules is nearly optimal and we propose to solve a parametric linear problem (given figure 2) similar to the limit problem stated before, in order to find the best one. If these assumptions are not satisfied, one can still solve a linear program, but we can not guarantee that the result will be nearly optimal.

We will assume, as in theorem 2 of [1], that there exists a generic schedule leading to an execution time linear in each parameter. Consider first that all

$$
\left\{
\begin{array}{l}
\forall i, 1 \leq i \leq k \\
X_i = X_i^{'} A_i \\
X_i = -X_i^{''} A_i \\
X_i^{'} \geq 0 \\
X_i^{''} \geq 0 \\
X_i D_i \geq 1 \\
\\
\forall i, 1 \leq i \leq k \\
\forall j \in Pred(i) \\
X_j D_{j,i} - X_i = X_{j,i} A_i \\
X_{j,i} \geq 0 \\
\forall t, 1 \leq t \leq s \\
-X_{j,i} b_{i,t} + c'_{i,t} - c'_{j,t} \geq 0 \\
-X_{j,i} b_i^* + X_j d_{j,i} + c_i^* - c_j^* \geq 1 \\
\\
\forall i, 1 \leq i \leq k \\
\forall t, 1 \leq t \leq s \\
\gamma_{1,t} - X_i' b_{i,t} - c'_{i,t} \geq 0 \\
\gamma_{2,t} - X_i'' b_{i,t} + c'_{j,t} \geq 0 \\
\gamma_1^* - X_i' b_i^* - c_i^* \geq 0 \\
\gamma_2^* - X_i'' b_i^* + c_j^* \geq 0 \\
\\
\min(n_1 \gamma_{1,1} + \cdots + n_s \gamma_{1,s} + \gamma_1^* + n_1 \gamma_{2,1} + \cdots + n_s \gamma_{2,s} + \gamma_2^*)
\end{array}
\right.
$$

Figure 2: Limit problem with affine dependences and s parameters

parameters except n_1 are fixed. We can then show that problem of figure 2 with $s = 1$ gives schedules nearly optimal in n_1 for each instance of the other parameters. Now, on this new problem, which is this time a **parametric** linear problem in n_1, we use the same technique for n_2, all other parameters being fixed. The solution is a conditional solution (n_1 be compared to n_2) but in each case, the solution is a polynomial in n_2 and a limit problem can be introduced. Actually, this problem is problem 2 with $s = 2$. By induction, this process is iterated to obtain problem 2 with all parameters in the objective function. Thus, arguments of the proof are very similar to those of theorem 2 of [1]. We can still apply a retiming technique to introduce at each step the c^* because on a cycle, we have:

$$-\sum_{cycle} X_{i,succ(i)} b_i + \sum_{cycle} X_i d_{i,succ(i)} \geq \text{cycle length}$$

where $b_i = n_1 b_{i,1} + \cdots + n_s b_{i,s} + b_i^*$. Thus, because all parameters are supposed non negative and because for all i and for all $t, 1 \leq t \leq s$, $X_{i,succ(i)} \geq 0$ and $b_{i,t} \geq 0$, all contributions in $-\sum_{cycle} X_{i,succ(i)} b_i$ are negative. Thus, at step one of the induction, we use the fact that:

$$-\sum_{cycle, i \geq 2} X_{i,succ(i)} b_i + X_{1,succ1} b_1^* + \sum_{cycle} X_i d_{i,succ(i)} \geq \text{cycle length}$$

to apply the retiming technique and then,

$$-\sum_{cycle, i \geq 3} X_{i,succ(i)} b_i + X_{1,succ1} b_1^* + X_{2,succ2} b_2^* + \sum_{cycle} X_i d_{i,succ(i)} \geq \text{cycle length}$$

and so on.

To conclude, note that the limit problem consists in transforming inequalities with parameters into several inequalities, each of them amounting to say that the components, factors of a given parameter, are nonnegative.

4 Example

We will present as an example the transformation of a program performing Gaussian elimination on a n-by-$(n + m)$ matrix. Here is the program:

```
for k=1 to n
{
    for i=k+1 to n
```

```
    {
        a(i,k)=a(i,k)/a(k,k) /* S1 */
        for j=k+1 to n+m
        {
            a(i,j)=a(i,j)-a(i,k)*a(k,j) /* S2 */
        }
    }
}
```

The domain of statement 1 is:

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} k \\ i \end{pmatrix} \leq n \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

The domain of statement 2 is:

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} k \\ i \\ j \end{pmatrix} \leq n \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + m \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ -1 \end{pmatrix}$$

There are affine dependences:

$$I_{2,1} \begin{pmatrix} k \\ i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} k \\ i \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$I_{1,2} \begin{pmatrix} k \\ i \\ j \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} k \\ i \\ j \end{pmatrix}$$

$$I_{2,2} \begin{pmatrix} k \\ i \\ j \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} k \\ i \\ j \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

and uniform dependences:

$$d_{2,2} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

The solution is:

```
X1 := array(1..2,[(1)=2,(2)=0]);
X2 := array(1..3,[(1)=2,(2)=0,(3)=0]);
c1_n := 0;
c2_m := 0;
c2_n := 0;
c2_m := 0;
c1 := 0;
c2 := 1;
temps := 2n;
```

Finally, our technique leads to the following program:

```
for k=1 to n
{
    doall i=k+1 to n
    {
        a(i,k)=a(i,k)/a(k,k)
    }

    doall i=k+1 to n
    {
        doall j=k+1 to n+m
        {
            a(i,j)=a(i,j)-a(k,j)*a(i,k)
        }
    }
}
```

References

[1] A.Darte and Y.Robert. Affine-by-statement scheduling of uniform loop nests over parametric domains. Technical Report 92-16, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, April 1992.