Optimal Parallel Schedules for Uniform Recurrence Equations

Xiaoxiong Zhong
University of Oregon

CIS-TR-92-12
April 1992

Abstract

We consider the problem of finding the optimal parallel schedule (also called **free schedule**) for a Uniform Recurrence Equation(URE) over an arbitrary family of integral polyhedra. Many computation structures such as nested loops with constant dependence vectors and systolic algorithms can be described by UREs. An optimal schedule is a function which schedules computations as soon as possible. We show that for a URE over a family of integral polyhedra, the free schedule is bounded within a constant to a piecewise quasi-linear function for computations not too "close" to the boundary, provided that the URE has a quasi-linear schedule. Furthermore, the piecewise quasi-linear function itself is a valid schedule and can be obtained by a parametric rational linear programming. We also show that it is always possible to choose a single quasi-linear schedule which is "almost" optimal with respect to the minimal execution time of the last computation over any domain in a family of bounded integral polyhedra. Intuitively, the result shows that we can effectively find a quasi-linear schedule which is "almost" optimal. This justifies using (quasi)-linear schedules in the parallelization of nested loops with constant dependence vectors and in the systolic array synthesis. The result is an extension to a result given in [KMW67].

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

1 Introduction

A Uniform Recurrence Equation(URE) describes a class of regular computation structures where computations are indexed by integral vectors over a convex polyhedron and performing a computation at an integral point needs (depends on) computation results at other integral points which have constant distances from its index. Such regular computation structures have many applications. For example, in the parallelization of nested loops, it can be used to describe a class of nested loops with constant dependence vectors [Lam74, Wol89]. In the design and synthesis of systolic arrays [Kun79], it is used to describe systolic algorithms [Qui84, RK86]. A *schedule* for a URE is a function which assigns time steps to computations based on the dependency constraints. This paper studies the problem of finding the optimal schedule for a URE over a family of *integral* polyhedra.

The pioneering work on UREs was done by Karp,Miller and Winograd [KMW67] who first studied computability and scheduling problems for such computations. Since then, their results have been used, modified and extended in many areas, especially in the two areas mentioned above [Wol89, Kun88]. The problem of finding the optimal schedule (called *free schedule* in [KMW67]) for a URE over an arbitrary integral convex polyhedron, however, is still open even though in [KMW67], it is proved that the free schedule of an arbitrary URE over a specific index domain (the first orthant of the infinite multi-dimensional grid region) is bounded within a constant to a rational linear function for computations not too "close" to the boundary.

In [Qui87], quasi-linear schedules are studied in systolic array design. In [FPP84], linear schedules for 25 common algorithms of nested loops with constant dependence vectors are studied and it is found that the difference between the optimal linear schedules and the optimal schedules is equal to one or zero for those nested loops. Extensive work on optimal *linear* schedules for UREs has been done by Shang and Fortes [SF89, SF91, Sha90]. Efficient algorithms are designed to find optimal linear schedules for both the total completion time and the completion time of a specific computation. The problem of finding the optimal schedule for a URE, however, is still open. This paper gives an answer to the problem.

2 Notations and Problem Definition

Throughout the paper, Z denotes the integer set and Q the rational number set. The integral grid is Z^n. Vectors are column vectors and are denoted as $\vec{I}$. For a vector $\vec{I} = (i_1, \ldots, i_n)^t$, $|\vec{I}| = (|i_1|, \ldots, |i_n|)^t$ and $\|\vec{I}\|_\infty = \max\{|i_1|, \ldots, |i_n|\}$. E_n is an $n \times n$ identity matrix. $\vec{1} = (1, \ldots, 1)^t$. A sequence of integers $a_1, \ldots, a_k$ is called *semipositive* if they are nonnegative but not all of them are zeros. For a rational polyhedron $\mathcal{P}$, its ***integer hull*** $\mathcal{P}_I$ is defined as the convex hull of the integral vectors in $\mathcal{P}$. $\mathcal{P}$ is called an *integral polyhedron* (IP) if $\mathcal{P}_I = \mathcal{P}$ ([Sch88]). Throughout this paper, we assume that $\mathcal{P}$ is an IP.

Definition 1 Let A be an $m \times n$ integral matrix, S_b be a subset of Z^m, a family of IPs $\mathcal{F}(A, S_b)$ is $\{\mathcal{P} | \mathcal{P} = \{\vec{I} | A\vec{I} \geq \vec{b}\}$ for any $\vec{b} \in S_b$ and $\mathcal{P}$ is an $IP\}$.

Intuitively, the "shape" of an IP is determined by its coefficient matrix A and $\mathcal{F}(A, S_b)$ is a collection of IPs which are of the same "shape". S_b is the range of the parameters. $\mathcal{P} = \{\vec{I} | A\vec{I} \geq \vec{b}\}$ can be decomposed as $\mathcal{P} = \mathcal{V} + \mathcal{C}$ where $\mathcal{V}$ is a polytope and $\mathcal{C} = \{\vec{y} | A\vec{y} \geq \vec{0}\}$ is the *characteristic* cone of $\mathcal{P}$ (cf. [Sch88]). Notice that $\mathcal{C}$ is independent of $\vec{b}$.

Property 1 *$\{I \in \mathcal{P}, I$ is integral$\} = \{e_1\vec{v}_1 + \ldots + e_q\vec{v}_q + f_1\vec{r}_1 + \ldots + f_p\vec{r}_p | e_i, f_j$ nonnegative and $e_1 + \ldots + e_p = 1\}$ for some integral vectors $\vec{v}_1, \ldots, \vec{v}_q$ and $\vec{r}_1, \ldots, \vec{r}_p$ where $\vec{r}_i s$ are independent of $\vec{b}$.*

Sketch of Proof: See [Sch88], page 234, formula (19). ∎

Since $\mathcal{P}$ is an IP, it is easy to see that $\mathcal{P} = \{e_1\vec{v}_1 + \ldots + e_q\vec{v}_q + f_1\vec{r}_1 + \ldots + f_p\vec{r}_p | e_i, f_j \geq 0$ and $e_1 + \ldots + e_q = 1\}$. Thus, *any* two $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{F}(A, S_b)$ have the same $\vec{r}_i$s. Throughout this paper, we assume $\mathcal{P}$ has such a representation $\mathcal{P} = \{V\vec{e} + R\vec{f} | \vec{e}, \vec{f} \geq \vec{0}$ & $e_1 + \ldots + e_q = 1\}$ where $V = (\vec{v}_1 \ldots \vec{v}_q), R = (\vec{r}_1 \ldots \vec{r}_p)$.

Definition 2 Let c be a function (computation) on Z^n, A be a $m \times n$ integral matrix, a **Uniform Recurrence Equation (URE)** $\mathcal{U}$ over a domain $\mathcal{P}$ in a family of $\mathcal{F}(A, S_b)$ is

$$c(\vec{I}) \quad = \quad f(c(\vec{I} - \vec{d}_1), \ldots, c(\vec{I} - \vec{d}_k))$$

where $\vec{I} \in \mathcal{P}$, $\vec{d}_1, \ldots, \vec{d}_k$ are n-dimensional *integral* vectors and f is an arbitrary function. $\vec{d}_i$s are called *dependency vectors* and $D = (\vec{d}_1, \ldots, \vec{d}_k)$ is called *dependency matrix*.

Thus, a URE $\mathcal{U}$, together with an IP $\mathcal{P}$, defines a computation structure. $\mathcal{U}$ and a family of IPs $\mathcal{F}(A, S_b)$, define a class of computation structures. A matrix multiplication example is given in Example 1 in the Appendix.

We say that $\vec{I} \in \mathcal{P}$ *depends* on $\vec{J} = \vec{I} - \vec{d}_i \in \mathcal{P}$ and denote it as $\vec{I} \rightarrow \vec{J}$. Intuitively, $\vec{I} \rightarrow \vec{J}$ means that $c(\vec{I})$ needs $c(\vec{J})$ as one of its arguments. Furthermore, if $\vec{I}_1 \rightarrow \vec{I}_2, \ldots, \vec{I}_n \rightarrow \vec{I}_{n+1}$ and $\vec{I}_1, \ldots, \vec{I}_{n+1} \in \mathcal{P}$, we denote $\vec{I}_1 \xrightarrow{n}_{\mathcal{P}} \vec{I}_{n+1}$. A *schedule* S is a function which maps an integral vector $\vec{I} \in \mathcal{P}$ to a positive integer such that if $\vec{I} \rightarrow \vec{J}$, then $S(\vec{I}) > S(\vec{J})$. Intuitively, a schedule is a function over $\mathcal{P}$ which schedules computation $c(\vec{I})$ at time $S(\vec{I})$. A URE is said to be *computable* in $\mathcal{F}(A, S_b)$ if there exists a schedule for any $\mathcal{P} \in \mathcal{F}(A, S_b)$. The maximum parallelism is achieved when the *free schedule* [KMW67] f is used to schedule the computations. The *free schedule* f is a schedule defined as

$$f(\vec{I}) \;=\; \begin{cases} 0, \text{ if no } \vec{J} \in \mathcal{P} \text{ such that } \vec{I} \rightarrow_{\mathcal{P}} \vec{J} \\ \max\{m | \vec{I} \xrightarrow{m}_{\mathcal{P}} \vec{J}, \vec{J} \in \mathcal{P}\} + 1, \text{otherwise} \end{cases}$$

The free schedule is the fastest schedule for every computation point in domain $\mathcal{P}$.

If a schedule S is of form $S(\vec{I}) = \lfloor \vec{s}^t \vec{I} + \alpha \rfloor$, it is called a *quasi-linear schedule*.

3 Computability of a URE

Definition 3 A family of IPs is said to be *extendible* to the dependency vectors $\vec{d}_1, \ldots, \vec{d}_k$ of URE $\mathcal{U}$ iff for any semipositive integers $l_1, \ldots, l_k$, there exists an IP $\mathcal{P}$ of the family such that it $\vec{I}_1 \xrightarrow{N}_{\mathcal{P}} \vec{I}_1 - \sum_{i=1}^{k} l_i \vec{d}_i$ for some integral point $\vec{I}_1 \in \mathcal{P}$ and some $N > 0$ (Note, N is not necessarily $\sum_{i=1}^{k} l_i$).

For example, the family of IPs for matrix multiplication in Example 1 in Appendix is extendible to the dependencies. In the sequel, we assume F is extendible to all the dependency vectors of URE $\mathcal{U}$.

Intuitively, a URE is computable iff there is no an infinite dependency chain.

Lemma 1 URE $\mathcal{U}$ is computable for a family $\mathcal{F}$ iff there are no semipositive integer sequence $a'_1, \ldots, a'_k$ such that $\sum_{i=1}^{k} a'_i \vec{d}_i = -\sum_{i=1}^{p} f'_i \vec{r}_i$ for some nonnegative integer f'_is.

Sketch of Proof: Suppose $\mathcal{U}$ is not computable over $\mathcal{P}$, then, there exists an infinite sequence of integral vector $\vec{I}_0 (= \vec{I}), \vec{I}_1, \ldots, \vec{I}_n, \ldots$ in $\mathcal{P}$ such that $\vec{I}_i \rightarrow_{\mathcal{P}} \vec{I}_{i+1}$ (hence, for $m < n$, $\vec{I}_m - \vec{I}_n = \sum_{i=1}^{k} a_i \vec{d}_i$ for some semipositive integers a_i's). Based on the representation of $\mathcal{P}$, for any $m \geq 0$, $\vec{I}_m = \sum_{i=1}^{q} e_i^m \vec{v}_i + \sum_{i=1}^{p} f_i^m \vec{r}_i$ for some nonnegative integers e_i^m's and f_i^m's and $= \sum_{i=1}^{q} e_i^m = 1$. Since there are only finitely many (2^q) different values of e_i^m's, there exists an infinite subsequence of $\vec{I}_1, \ldots, \vec{I}_n, \ldots$ such that their e_i^m's are repetitive. Among this subsequence, we *can* further choose an infinite subsequence whose f_i^m are *nondecreasing* for each $i = 1, \ldots, p$. Let $\vec{I}_m$ and $\vec{I}_n$ are two from this subsequence ($m < n$), we have $\vec{I}_m - \vec{I}_n = \sum_{i=1}^{k} a'_i \vec{d}_i = -\sum_{i=1}^{p} f'_i$ for some semipositive integers a'_i and nonnegative integers f'_i.

Conversely, if there are semipositive integer sequence $a_1, \ldots, a_k$ such that $\sum_{i=1}^{k} a_i \vec{d}_i = -\sum_{i=1}^{p} f_i \vec{r}_i$ for some nonnegative integer f_i's. Since $\mathcal{F}$ is extendible to $\vec{d}_i$'s, there exists a $\mathcal{P} \in \mathcal{F}$ and an integral $\vec{I} \in \mathcal{P}$ such that $\vec{I}_0 (= \vec{I}) \rightarrow_{\mathcal{P}} \vec{I}_1 \rightarrow_{\mathcal{P}} \ldots, \vec{I}_N (= \vec{I} - \sum_{i=1}^{k} a_i \vec{d}_i)$. But $\vec{I}_i + \sum_{i=1}^{p} f_i \vec{r}_i \in \mathcal{P}$ (since $\vec{I}_i \in \mathcal{P}$ and recall $\mathcal{P}$'s representation), we have an infinite dependency chain $\vec{I}_0 \rightarrow_{\mathcal{P}} \vec{I}_1 \rightarrow_{\mathcal{P}} \ldots \vec{I}_N \rightarrow_{\mathcal{P}} \vec{I}_1 + \sum_{i=1}^{p} f_i \vec{r}_i \ldots$. Thus, $\mathcal{U}$ is not computable over $\mathcal{P}$. ∎

The above result is a generalization to a result given in [KMW67] where a similar condition is given for a URE over a specific domain (the first orthant of the gird).

Theorem 1 $\mathcal{U}$ is computable iff there exists an n-dimensional vector s such that

$$\begin{array}{l} \vec{s}^t\vec{d}_i \geq 1, i = 1, \ldots, k \\ \vec{s}^t\vec{r}_i \geq 0, i = 1, \ldots, p \end{array} \qquad (1)$$

Sketch of Proof: From Lemma 1, $\mathcal{U}$ is computable iff linear programming (LP) $\max\{\sum_{i=1}^k a_i | \sum_{i=1}^k a_i\vec{d}_i + \sum_{i=1}^p f_i\vec{r}_i = 0, a_i, f_j \geq 0\}$ has a finite optimal value 0, iff its dual problem $\min\{0|\vec{s}^t\vec{d}_i \geq 1, \vec{s}^t\vec{r}_i \geq 0\}$ has a feasible solution. ▮

Corollary 1 *$\mathcal{U}$ is computable iff there exists a quasi-linear schedule for each $\mathcal{P}$.*

Sketch of Proof: For a $\vec{s}$ which satisfying condition (1), we can always find a constant α such that $\vec{s}^t\vec{v}_i + \alpha \geq 0$ for all $\vec{v}_i$ in an IP $\mathcal{P}$. It is easy to prove that $L(\vec{I}) = \lfloor \vec{s}^t\vec{I} + \alpha \rfloor$ is a valid schedule for $\mathcal{U}$ on $\mathcal{P}$ (simply check $L(\vec{I}) \geq 0$ and $L(\vec{I}_1) > L(\vec{I}_2)$ if $\vec{I}_1 \rightarrow \vec{I}_2$). ▮

Condition 1 is similar to a result given by Quinton ([Qui87]). When the family consists of only bounded (finite) IPs (i.e., $p = 0$), this is called "separating hyperplane" ([KMW67, Qui87]): $\vec{s}$ can be thought of as the norm of a hyperplane which separates dependency vectors from the first orthant.

4 The Free Schedule

We first consider the following two *rational* linear programming problems for an *integral* vector $\vec{I} \in \mathcal{P}$.

$$\left\{ \begin{array}{l} m_1(\vec{I}) = \max \sum_{i=1}^k u_i \\ \text{subject to} \\ 1)\ u_i \geq 0, i = 1, \ldots, k \\ 2)\ \vec{I} - \sum_{i=1}^k u_i\vec{d}_i \in \mathcal{P} \end{array} \right. \quad (\mathrm{I}) \qquad \left\{ \begin{array}{l} m_2(\vec{I}) = \min \vec{\lambda}^t(A\vec{I} - \vec{b}) \\ \text{subject to} \\ 1)\ \lambda_i \geq 0, i = 1, \ldots, k \\ 2)\ \vec{\lambda}^t A\vec{d}_i \geq 1, i = 1, \ldots, k \end{array} \right. \quad (\mathrm{II})$$

Lemma 2 If URE $\mathcal{U}$ is computable, then both (I) and (II) have a common, finite optimal value $m(\vec{I})$ for an integral vector $\vec{I} \in \mathcal{P}$.

Proof: Based on the representation of $\mathcal{P}$, the feasible region of (I) can be rewritten as follows

$$\left\{ \begin{array}{l} m_1(\vec{I}) = \max \sum_{i=1}^k u_i \\ \text{subject to} \\ 1)\ \vec{u}, \vec{e}, \vec{f} \geq \vec{0} \\ 2)\ \vec{I} - D\vec{u} - V\vec{e} - R\vec{f} = \vec{0} \\ 3)\ \vec{1}^t\vec{e} = 1 \end{array} \right.$$

Its dual problem is

$$\min\{(\vec{s}^t \ \alpha)\begin{pmatrix} \vec{I} \\ 1 \end{pmatrix} | (\vec{s}^t \ \alpha)\begin{pmatrix} D & R & V \\ 0 & 0 & 1 \end{pmatrix} \geq (\vec{1}^t_k \ \vec{0}^t_{p+q})\} \text{ (III)}$$

where α is a scalar variable and $\vec{1}^t_k$ ($\vec{0}^t_{p+q}$) is a k-dimensional ($(p+q)$-dimensional) vector with all 1 (0) components. Based on Corollary 1, the feasible region of (III) is nonempty if $\mathcal{U}$ is computable (we can always choose α to satisfy $\vec{s}^t\vec{v}_i + \alpha \geq 0$ for all $i = 1 \ldots, q$). Hence (III) has a feasible solution. But the feasible region of (I) is not empty since $\vec{u} = \vec{0}$ is a feasible solution. Hence, by duality theorem, (I) has a finite maximum. Therefore, its dual problem (II) must have a finite minimum too. Thus, $m_1(\vec{I}) = m_2(\vec{I})$ and they are denoted as $m(\vec{I})$. ▮

Notice that the feasible region of (II) is independent of $\vec{b}$. The objective function of (II) is a linear function of $\vec{I}$ and $\vec{b}$. Based on a property in parametric linear programming(see, for example, page 15 in [Nau77]), we can show that $m(\vec{I})$ is a piece-wise linear function of $\vec{I}$ and $\vec{b}$. Furthermore, for a fixed $\mathcal{P}$ (i.e., $\vec{b}$ is fixed), we know that $m(\vec{I})$ is also a piece-wise linear function of $\vec{I}$ (a projection of a piecewise linear function is still a piecewise linear function).

Lemma 3 Integer function $m'(\vec{I}) = \lfloor m(\vec{I}) \rfloor$ is a valid schedule for URE $\mathcal{U}$ in $\mathcal{P}$.

Proof: Let $\vec{I}_1, \vec{I}_2 \in \mathcal{P}$ and $\vec{I}_1 \ \vec{N}_{\mathcal{P}} \ \vec{I}_2$ for some $N = \sum_{i=1}^k u_i > 0$ where $u_i, i = 1, \ldots, k$ are nonnegative integers. $\vec{I}_2 = \vec{I}_1 - \sum_{i=1}^k u_i\vec{d}_i$. Denote $D_{\vec{I}_2} = \{\vec{u}' = (u'_1, \ldots, u'_k) | \vec{I}_2 - \sum_{i=1}^k u'_i\vec{d}_i \in \mathcal{P}\}$. For any $\vec{u}' \in D_{\vec{I}_2}$, we have $\vec{I}_1 - \sum_{i=1}^k (u_i + u'_i)d_i \in \mathcal{P}$. Thus, $m(\vec{I}_1) \geq \max_{\vec{u}' \in D_{\vec{I}_2}} \sum_{i=1}^k (u_i + u'_i) = \max_{\vec{u}' \in D_{\vec{I}_2}} \sum_{i=1}^k (u'_i) + N$. But $m(\vec{I}_2) = \max_{u' \in D_{\vec{I}_2}} \sum_{i=1}^k u'_i$ and since $N > 0$, we conclude $\lfloor m(\vec{I}_1) \rfloor \geq \lfloor m(\vec{I}_2) + N \rfloor > \lfloor m(\vec{I}_2) \rfloor$. ▮

We call $m'(\vec{I})$ a piecewise quasi-linear schedule. One may attempt to conjecture that $m'(\vec{I})$ is the free schedule for the URE. However, this is not correct. The main reason is that for $\vec{I}_1, \vec{I}_2 \in \mathcal{D}$ such that $\vec{I}_2 = \vec{I}_1 - \sum_{i=1}^k u_i\vec{d}_i$, it is not necessarily true that $\vec{I}_1 \xrightarrow{N}_{\mathcal{P}} \vec{I}_2$ for $N = \sum_{i=1}^k u_i$. This is because the dependencies from $\vec{I}_1$ to $\vec{I}_2$ may first go out of $\mathcal{P}$ and then go back to $\mathcal{P}$ later. Example 2 in Appendix shows this.

However, we are able to prove that for computations which are not too "close" to the boundary, the difference between $m'(\vec{I})$ and $f(\vec{I})$ (f, the free schedule) is bounded within a constant which is independent of $\vec{b}$. The method used is similar to that in [KMW67]. We first define another polyhedron $\mathcal{P}_s$ for computation points which are not too "close" to the boundary of $\mathcal{P}$ as $\mathcal{P}_s = \{I | A\vec{I} \geq \vec{b} + \vec{\pi}\}$ where

$$\vec{\pi} = \sum_{i=1}^k |A\vec{d}_i|.$$

Since $\vec{\pi} \geq \vec{0}$, $\mathcal{P}_s \subseteq \mathcal{P}$. We assume that $\mathcal{P}_s$ is not empty. We first prove the following lemma.

Lemma 4 If $\vec{I}_1, \vec{I}_2 = \vec{I}_1 - \sum_{i=1}^k u_i\vec{d}_i \in \mathcal{P}_s$ for some semipositive integers $u_1, \ldots, u_k$, then $\vec{I}_1 \overset{N}{\rightarrow}_{\mathcal{P}} \vec{I}_2$ where $N = \sum_{i=1}^k u_i$.

Proof: We prove the following claims successively.

Claim 1: If $\vec{\delta} \in \mathcal{P}_s$ and $\theta_i \in \{0, 1\}, i = 1, \ldots, k$, then $\vec{r} = \vec{\delta} + \sum_{i=1}^k \theta_i\vec{d}_i \in \mathcal{P}$.

Proof of Claim 1: Simply check

$$\begin{aligned}
A\vec{r} &= A\vec{\delta} + \sum_{i=1}^k \theta_i A\vec{d}_i = A\vec{\delta} - \vec{\pi} + (\sum_{i=1}^k \theta_i A\vec{d}_i + \vec{\pi}) \\
&= A\vec{\delta} - \vec{\pi} + (\sum_{i=1}^k \theta_i A\vec{d}_i + \sum_{i=1}^k |A\vec{d}_i|) = A\vec{\delta} - \vec{\pi} + \sum_{i=1}^k (\theta_i A\vec{d}_i + |A\vec{d}_i|) \\
&\geq A\vec{\delta} - \vec{\pi} \geq \vec{b}
\end{aligned}$$

Claim 2: Let $\vec{r}$ be the one defined in **Claim 1** and it is integral, $\vec{r} - \sum_{i=1}^k \alpha_i\vec{d}_i = \vec{r}'$ where $\alpha_i \in \{0, 1\}$, then $\vec{r} \overset{M}{\rightarrow}_{\mathcal{P}} \vec{r}'$ where $M = \sum_{i=1}^k \alpha_i$.

Proof of Claim 2: Without loss of generality, we assume that $\alpha_i = 1, i = 1, \ldots, L$ and $\alpha_i = 0, i = L + 1, \ldots, k$ for some $L \leq k$. Let $\vec{r}'_j = \vec{r} - \sum_{i=1}^j \vec{d}_i, j = 0, 1, \ldots, L$ (hence, $\vec{r} = \vec{r}'_0$ and $\vec{r}' = \vec{r}'_L$), we have

$$\begin{aligned}
A\vec{r}'_j &= A\vec{r} - \sum_{i=1}^j A\vec{d}_i = A(\vec{\delta} + \sum_{i=1}^k \theta_i\vec{d}_i) - \sum_{i=1}^j A\vec{d}_i \\
&= A\vec{\delta} + \sum_{i=1}^k \theta_i A\vec{d}_i - \sum_{i=1}^j A\vec{d}_i = A\vec{\delta} + \sum_{i=1}^j (\theta_i - 1)A\vec{d}_i + \sum_{i=j+1}^k \theta_i A\vec{d}_i \\
&= A\vec{\delta} - \vec{\pi} + \sum_{i=1}^j (|A\vec{d}_i| + (\theta_i - 1)(A\vec{d}_i)) + \sum_{i=j+1}^k (|A\vec{d}_i| + \theta_i A\vec{d}_i) \geq A\vec{\delta} - \vec{\pi} \\
&\geq \vec{b}
\end{aligned}$$

Hence, $\vec{r}'_j \in \mathcal{P}$. Therefore, $\vec{r} = \vec{r}'_0 \rightarrow_{\mathcal{P}} \vec{r}'_1 \rightarrow_{\mathcal{P}} \vec{r}'_2 \ldots \rightarrow_{\mathcal{P}} \vec{r}'_L = \vec{r}'$, we prove the claim.

Proof of Lemma 4: Let

$$\vec{r}_c = \vec{I}_1 - \sum_{i=1}^k \lfloor \frac{cu_i}{N} \rfloor \vec{d}_i, c = 0, 1, \ldots, N.$$

$\vec{r}_0 = \vec{I}_1, \vec{r}_N = \vec{I}_2$. Rewrite $\vec{r}_c$ as follows

$$\vec{r}_c = \vec{I}_1 - \frac{c}{N}\sum_{i=1}^k u_i\vec{d}_i + \sum_{i=1}^k (\frac{cu_i}{N} - \lfloor \frac{cu_i}{N} \rfloor)\vec{d}_i$$

Since $\mathcal{P}_s$ is a convex polyhedron,

$$\vec{I_1} - \frac{c}{N}\sum_{i=1}^{k} u_i\vec{d_i} = (1-\frac{c}{N})\vec{I_1} + \frac{c}{N}\vec{I_2} \in \mathcal{P}_s.$$

Based on **Claim 1**, $\vec{r_c} \in \mathcal{P}, c = 0, 1, \ldots, N$. Furthermore, notice that

$$\vec{r}_{c+1} = \vec{r_c} - \sum_{i=1}^{k}(\lfloor\frac{(c+1)u_i}{N}\rfloor - \lfloor\frac{cu_i}{N}\rfloor)\vec{d_i},$$

and

$$\lfloor\frac{(c+1)u_i}{N}\rfloor - \lfloor\frac{cu_i}{N}\rfloor \in \{0,1\}.$$

Based on **Claim 2**, $\vec{r_c} \xrightarrow{L_c}_{\mathcal{P}} \vec{r}_{c+1}$ where

$$L_c = \sum_{i=1}^{k}(\lfloor\frac{(c+1)u_i}{N}\rfloor - \lfloor\frac{cu_i}{N}\rfloor).$$

But $\sum_{c=0}^{N-1} L_c = N$, hence we prove the lemma. ∎

Theorem 2 For a family of CPDs $\mathcal{F}(A, S_b)$, there exists a constant C such that for every $\mathcal{P} \in \mathcal{F}(A, S_b)$, $m'(\vec{I}) - f(\vec{I}) \leq C$ for any integer vector $I \in \mathcal{P}_s$.

Proof: Consider the following rational linear programmings for any integral vector $I \in \mathcal{P}_s$.

$$\left\{\begin{array}{l} M_1(\vec{I}) = \max\sum_{i=1}^{k} u_i \\ \text{subject to} \\ 1)\ u_i \in Q \text{ and } u_i \geq 0, i = 1,\ldots,k \\ 2)\ I - \sum_{i=1}^{k} u_i\vec{d_i} \in \mathcal{P}_s \end{array}\right. \quad \text{(I')} \qquad \left\{\begin{array}{l} M_2(\vec{I}) = \min \lambda^t(A\vec{I} - (\vec{b} + \vec{\pi})) \\ \text{subject to} \\ 1)\ \lambda_i \geq 0, i = 1,\ldots,k \\ 2)\ \vec{\lambda}^t A\vec{d_i} \geq 1, i = 1,\ldots,k \end{array}\right. \quad \text{(II')}$$

$M_1(\vec{I})$ is finite since $u_i = 0$ is a feasible solution and $M_1(\vec{I}) \leq m(\vec{I})$ (since $\mathcal{P}_s \subseteq \mathcal{P}$). Thus, both (I') and (II') have a common optimal value $M(\vec{I}) = M_1(\vec{I}) = M_2(\vec{I})$. Let the optimal solution of the *integer* linear programming by restricting u_is to be integers in (I') be $M'(\vec{I})$, based on a result in integer linear programming(see, page 239-240, theorem 17.2, in [Sch88]), there exists a constant C_1 which is *independent* of $\vec{b}$ such that $0 \leq M(\vec{I}) - M'(\vec{I}) \leq C_1$. Now, consider the difference between the free schedule $f(\vec{I})$ of $\vec{I}$ in $\mathcal{P}_s$ and $M'(\vec{I})$. Since $M'(\vec{I}) = N = \sum_i^k u_i$ for some integer $u_i \geq 0$ such that $\vec{I'} = \vec{I} - \sum_i^k u_i\vec{d_i} \in \mathcal{P}_s$, from Lemma 4, we have $\vec{I} \xrightarrow{N}_{\mathcal{P}} \vec{I'}$. Thus, $f(\vec{I}) \geq N(= M'(\vec{I}))$. Hence,

$$m'(\vec{I}) - f(\vec{I}) \leq m'(\vec{I}) - M'(\vec{I}) \leq m(\vec{I}) - M'(\vec{I}) \leq m(\vec{I}) - M(\vec{I}) + C_1$$

Notice that (II) and (II') have optimal solutions and both of their feasible regions are the same (denoted as $\Delta = \{\vec{\lambda} | \vec{\lambda}^t A\vec{d_i} \geq 1, i = 1,\ldots,k, \vec{\lambda} \geq \vec{0}\}$). There exists

$\vec{\lambda}' \in \Delta$ such that

$$M(\vec{I}) = \vec{\lambda}'^t(A\vec{I} - (\vec{b} + \vec{\pi})).$$

But

$$m(\vec{I}) \leq \vec{\lambda}'^t(A\vec{I} - \vec{b}),$$

we have

$$m(\vec{I}) - M(\vec{I}) \leq \vec{\lambda}'^t\vec{\pi} \leq \max_{\vec{\lambda}\in\Delta} \vec{\lambda}^t\vec{\pi}$$

Hence, letting $C = C_1 + \max_{\vec{\lambda}\in\Delta} \vec{\lambda}^t\vec{\pi}$, we prove the theorem. ∎

In [KMW67], Karp et al. show an example where for some points "close" to the boundary, the difference between $m(\vec{I})$ and $f(\vec{I})$ are not bounded to a constant for the first orthant of n-dimensional grid. By slightly modifying their example, we can also show that even for a family of bounded IPs, difference between $m(\vec{I})$ and $f(\vec{I})$ may still be unbounded. This is given in Example 2 in the Appendix.

5 The Optimal Schedule for The Last Computation

In many applications, it is often desired to find a schedule to minimize the completion time of the *whole* computation. This problem is meaningful only if the domain $\mathcal{P}$ is bounded (i.e., a polytope). Thus, in this section, $\mathcal{P}$ is assumed to be an integral polytope. A computation $c(\vec{I})$ is called a last computation in domain $\mathcal{P}$ if there is no other point $\vec{I}' \in \mathcal{P}$ such that $\vec{I}' \xrightarrow{N}_{\mathcal{P}} \vec{I}$ for some $N > 0$. It is possible that there are more than one last computations. A schedule which minimizes the completion time of the whole computation is the one which schedules all the last computations at the time given by the free schedule. In this section, we show that if URE $\mathcal{U}$ has only a single last computation $c(\vec{I_l})$, then there exists a *single* quasi-linear schedule which minimizes the completion time of the whole computation (i.e., the time for $\vec{I_l}$).

Since the last computation point $\vec{I_l}$ usually lies close to the boundary (or in the boundary), Thm. 2 is not directly applicable for computation at $\vec{I_l}$. In the following, however, we prove that $m'(\vec{I_l})$ is still bounded to $f(\vec{I_l})$ within a constant independent of $\vec{b}$.

Lemma 5 For any $\vec{I} \in \mathcal{P}_s$, $f(\vec{I_l}) \geq M'(\vec{I})$ where $M'(\vec{I})$ is defined in the proof of Thm. 2.

Proof: Since $\vec{I_l}$ is the only last computation point in $\mathcal{P}$ and $\mathcal{P}$ is a finite polytope, for any $\vec{I} \in \mathcal{P}_s$, $\vec{I_l} \xrightarrow{L}_{\mathcal{P}} I$ for some integer $L > 0$. This implies $f(\vec{I_l}) \geq f(\vec{I})$. Let $u_1, \ldots, u_k \geq 0$ be the optimum solution for $M'(\vec{I})$. Based on Lemma 4, $\vec{I} \xrightarrow{N}_{\mathcal{P}} \vec{I} - \sum_{i=1}^{k} u_i\vec{d_i}$ where $N = M'(\vec{I}) = \sum_{i=1}^{k} u_i$. It follows that $f(\vec{I}) \geq M'(\vec{I})$. Hence, $f(\vec{I_l}) \geq M'(\vec{I})$. ∎

Lemma 6 For any $\vec{I} \in \mathcal{P}_s$, there exists a constant K_1 such that $m(\vec{I}) - M'(\vec{I}) \leq K_1$.

Proof: We prove that $m(\vec{I}) - M(\vec{I}) \leq K'$ for some constant K' first.

Since $m(\vec{I}) = \max\{\vec{1}^t\vec{u} | \begin{pmatrix} AD \\ -E_k \end{pmatrix} \vec{u} \leq \begin{pmatrix} A\vec{I} - \vec{b} \\ 0 \end{pmatrix}\}$ and $M(\vec{I}) = \max\{\vec{1}u | \begin{pmatrix} AD \\ -E_k \end{pmatrix} \vec{u} \leq \begin{pmatrix} A\vec{I} - \vec{b} - \vec{\pi} \\ 0 \end{pmatrix}\}$, based on a well known result on the sensitivity analysis in linear programming (see, Thm.10.5, page 126 in [Sch88]), for the optimum solution $\vec{u}'$ of $m(\vec{I})$ and the optimum solution $\vec{u}''$ of $M(\vec{I})$, $\|\vec{u}' - \vec{u}''\|_\infty \leq n\beta\| \begin{pmatrix} A\vec{I} - \vec{b} \\ 0 \end{pmatrix} - \begin{pmatrix} A\vec{I} - \vec{b} - \vec{\pi} \\ 0 \end{pmatrix} \|_\infty$ where β is a constant chosen such that all of the entries in B^{-1} for each nonsingular submatrix B of $\begin{pmatrix} AD \\ -E_k \end{pmatrix}$ is at most β. Hence, there exists a constant K' which is independent of $\vec{b}$ such that $m(\vec{I}) - M(\vec{I}) = \vec{1}^t(\vec{u}' - \vec{u}'') \leq K'$.

Since $M(\vec{I}) - M'(\vec{I}) \leq C_1$, $m(\vec{I}) - M'(\vec{I}) \leq m(\vec{I}) - M(\vec{I}) + C_1 \leq K' + C_1$. Let $K_1 = K' + C_1$. We prove the lemma. ∎

Lemma 7 There exists an $\vec{I}_0 \in \mathcal{P}_s$ such that $m(\vec{I}_l) - m(\vec{I}_0) \leq K_2$ for some constant K_2 which is independent of $\vec{b}$.

Proof: Consider two linear programming problems, $\max\{\vec{0}^t\vec{I} | (-A)\vec{I} \leq -\vec{b}\}$ and $\max\{\vec{0}^t\vec{I} | (-A)\vec{I} \leq -(\vec{b} + \vec{\pi})\}$. $\vec{I}_l$ is a feasible and optimum solution to the first problem. Based on similar argument in the proof of Lemma 6, there exists an optimum (feasible) solution $\vec{I}_0$ to the second problem such that $\|\vec{I}_l - \vec{I}_0\|_\infty \leq K' = n\beta\|\vec{\pi}\|_\infty$ where β is a constant independent of $\vec{b}$. For $\vec{I}_0$, there exists a $\vec{\lambda}' \in \Delta$ where Δ is the feasible region of (II') such that $m(\vec{I}_0) = \vec{\lambda}'^t(A\vec{I}_0 - \vec{b} - \vec{\pi})$. But $m(\vec{I}_l) \leq \vec{\lambda}'^t(A\vec{I}_l - \vec{b})$, we have $m(\vec{I}_l) - m(\vec{I}_0) \leq \vec{\lambda}'^t(A(\vec{I}_l - \vec{I}_0) + \vec{\pi})$. But $\|\vec{I}_l - \vec{I}_0\|_\infty \leq K'$, we easily know that there is a constant K_2 such that $m(\vec{I}_l) - m(\vec{I}_0) \leq K_2$. ∎

Theorem 3 If $\vec{I}_l$ is the only last computation point in $\mathcal{P}$ for URE $\mathcal{U}$, then there exists a constant K independent of $\vec{b}$ such that $m'(\vec{I}_l) - f(\vec{I}_l) \leq K$.

Proof: From Lemma 5 and Lemma 6, we have

$$\begin{aligned} m'(\vec{I}_l) - f(\vec{I}_l) &\leq m(\vec{I}_l) - \max_{\vec{I} \in \mathcal{P}_s}\{M'(\vec{I})\} = \min_{\vec{I} \in \mathcal{P}_s}\{m(\vec{I}_l) - M'(\vec{I})\} \\ &= \min_{\vec{I} \in \mathcal{P}_s}\{m(\vec{I}_l) - m(\vec{I}) + m(\vec{I}) - M'(\vec{I})\} \leq \min_{\vec{I} \in \mathcal{P}_s}\{m(\vec{I}_l) - m(\vec{I})\} + K_1 \end{aligned}$$

Based on Lemma 7, there exists an $\vec{I}_0 \in \mathcal{P}_s$ such that $\min_{\vec{I} \in \mathcal{P}_s}\{m'(\vec{I}_l) - m(\vec{I})\} \leq m(\vec{I}_l) - m(\vec{I}_0) \leq K_2$, letting $K = K_1 + K_2$, we prove the theorem. ∎

Therefore, we can first find an optimum solution $\vec{\lambda}$ and α for $\vec{I}_l$ in linear programming (III) which minimizes $m(\vec{I}_l)$. Based on Corollary 1, quasi-linear function $L(\vec{I}) = \lfloor \vec{\lambda}^t I + \alpha \rfloor$ is also a schedule. Since $L(\vec{I}_l) = m'(\vec{I}_l)$, $L(\vec{I})$ minimizes the execution time of computation at $\vec{I}_l$ and thus minimizes the completion time of the whole computation.

6 Remarks

We make the following remarks on the results presented above.

- In this paper, we consider UREs over a *parameterized* domain (or a family of domains). This is important as the domains (iteration spaces) of many applications (nested loops, systolic algorithms) belong to a family, not a single instance. In fact, for a *single finite* polyhedron, the problems considered here become trivial. Our results can be used in a parallel compiler as at *compile time*, the parametric $m'(I)$ can be solved, which will help the compiler in scheduling and partitioning of the nested loops.

- In this paper, we only consider UREs over a family of *integral* polyhedra. This, however, is still applicable to most of applications. For example, the loop bounds of many nested loops are integers and hence the vertices of their iteration spaces are integral, which implies that the iteration spaces are IPs (as long as all the vertices of a finite polyhedron are integral, then it is an IP).

- In Sec. 3, we derived an iff condition for the computability of a URE based on domain extendible condition. Again, we believe that this condition is not too strict. A typical family of domains which is extendible to any dependency vector is a hypercube. Moreover, the rest of our results can be easily derived from the existence of a quasi-linear schedule for each domain in the family (i.e., condition (1) in Sec. 3, instead of the computability condition we derived).

- The constants we derived to bound the schedule $m'(\vec{I})$ in fact are exponential to the dimension of the domain (i.e., n). However, In most applications, n is usually rather small (3,4 or 5).

An important open problem is to extend our results for finding a single schedule which minimizes the whole computation for the case where there are more than one last computation points.

Appendix

THIS SECTION SERVES AS REFERENCE. TWO EXAMPLES ARE GIVEN HERE

Example 1: Matrix multiplication $Z = X \times Y$ can be described by a URE over a family of domains $\mathcal{F}(A, S_b)$ with $A = (-E_3, E_3)^t$ and $S_b = \{(-N, -N, -N, 1, 1, 1)^t, N \geq 1\}$. The URE, over a domain $\mathcal{P} = \{(i, j, k)^t | 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq k \leq N\}$ is for all $(i, j, k)^t \in \mathcal{P}$

$$\left\{ \begin{array}{lcl} x(i,j,k) & = & x(i,j-1,k) \\ y(i,j,k) & = & y(i-1,j,k) \\ z(i,j,k) & = & z(i,j,k-1) + x(i,j-1,k) \times y(i-1,j,k) \end{array} \right.$$

and the boundary conditions are

$$\left\{ \begin{array}{lcl} x(i,0,k) & = & X_{ik} \\ y(0,j,k) & = & Y_{kj} \\ z(i,j,0) & = & 0 \\ Z_{ij} & = & z(i,j,N) \end{array} \right.$$

Example 2: Consider a URE with three dependency vectors $\vec{d_1} = (-1, 1, 1)^t, \vec{d_2} = (1, -1, 1)^t$ and $\vec{d_3} = (0, 0, 2)$ over a family $\mathcal{F}(A, S_b)$ of $2N \times 2N \times 2N$ cubes (for $N \geq 1$). For $\vec{I} = (1, 1, 2N)^t$, $\vec{I} - (N-1)\vec{d_1} - (N-1)\vec{d_2} = (1, 1, 2)^t$. $m(\vec{I}) = 2N - 2$. But $f(\vec{I}) = N - 1$. Thus $m(\vec{I}) - f(\vec{I}) = N - 1$ which can not be bounded to a constant independent of the size of the cube.

References

[FPP84] J.A.B. Fortes and F. Parisi-Presicce. Optimal linear schedule for the parallel execution of algorithms. In *Proc. of 1984 Int'l Conference on Parallel Processing*, 1984.

[KMW67] R. M. Karp, R. E. Miller, and S. Winograd. The organization of computations for uniform recurrence equations. *JACM*, 14(3):563–590, July 1967.

[Kun79] H. T. Kung. Let's design algorithms for VLSI. In *Proc. Caltech Conference on VLSI*, January 1979.

[Kun88] S. Y. Kung. *VLSI Array Processors*. Prentice Hall, 1988.

[Lam74] Leslie Lamport. The parallel execution of DO loops. *Communications of the ACM*, pages 83–93, February 1974.

[Nau77] R.M. Nauss. *Parametric Integer Programming*. University of Missouri Press, Columbia, Missouri, 1977.

[Qui84] Patrice Quinton. Automatic generation of systolic arrays from uniform recurrent equations. In *Proc. 11 th Annu. Int. Symp. Comput. Architecture*, June 1984.

[Qui87] Patrice Quinton. *in Automata Networks in Computer Science*, chapter 9: The Systematic Design of Systolic Arrays, pages 229–260. Princeton University Press, 1987. Preliminary versions appear as IRISA Tech Reports 193 and 216, 1983.

[RK86] Sailesh Rao and Thomas Kailath. What is a systolic algorithm. In *Proceedings, Highly Parallel Signal Processing Architectures*, pages 34–48, Los Angeles, Ca, Jan 1986. SPIE.

[Sch88] A. Schrijver. *Theory of Integer and Linear Programming.* John Wiley and Sons, 1988.

[SF89] W. Shang and J. A. B. Fortes. On the optimality of linear schedules. *Journal of VLSI Signal Processing*, 1:209–220, 1989.

[SF91] W. Shang and J.A.B. Fortes. Time optimal linear schedules for algorithms with uniform dependencies. *IEEE Trans. on Computers*, 40(6), June 1991.

[Sha90] W. Shang. *Scheduling, partitioning and mapping of uniform dependence algorithms on processor arrays.* PhD thesis, Purdue University, W. Lafayette, IN, May 1990.

[Wol89] M. Wolfe. *Optimizing Supercompiler for Supercomputers.* Pitman, London, 1989.