Feedback Guided Dynamic Loop Scheduling; A Theoretical Approach

Tatiana Tabirca
University of Manchester
Department of Computer Science
Oxford Road, Manchester, M13 9NG, UK
tabircat@cs.man.ac.uk

Len Freeman
University of Manchester
Department of Computer Science
Oxford Road, Manchester, M13 9NG, UK
freeman@cs.man.ac.uk

Sabin Tabirca
University College Cork
Department of Computer Science
College Road, Cork, Ireland
s.tabirca@cs.ucc.ie

Laurence Tianruo Yang
St. Francis Xavier University
Department of Computer Science
P.O.Box 5000, Antigonish, B2G 2W5, Canada
lyang@stfx.ca

Abstract

In this paper we review existing loop scheduling algorithms and also describe the feedback-guided dynamic loop scheduling (FGDLS) algorithm that was proposed in Bull et al. [2] and Bull [1]. The FGDLS algorithm uses a feedback mechanism to schedule a parallel loop within a sequential outer loop. It has been shown to perform well for scheduling problems for which the load associated with the parallel loop changes relatively slowly as the outer sequential loop executes. However the question of convergence of the FGDLS algorithm has remained an open question. In this paper we are able to establish sufficient conditions (essentially requiring that the workload does not change too rapidly with loop iteration count) for the (global) convergence of a continuous analogue of the feedback-guided algorithm.

1 Introduction

Load imbalance is observed to be a significant overhead in many parallel implementations. Further, since loops are the dominant source of parallelism for many applications, a variety of algorithms that aim to schedule loop iterations to processors of a shared-memory machine in an almost optimal way (so-called loop scheduling algorithms) have been designed.

Many existing loop scheduling algorithms are based on guided self-scheduling (Polychronopoulos and Kuck [8]) or some variant of guided self-scheduling (see for example Eager and Zahorjan [3], Hummel et al. [4], Lucco [6], Tzen and Ni [11]). These algorithms proceed by dividing the loop iteration into a large number of chunks (typically significantly more than the number of processors, and the chunks are of decreasing size as the iteration progresses) and the chunks are assigned to processors. One of the motivations for this approach is the assumption is that each execution of a loop is independent of any previous executions of the same loop, and therefore has to be rescheduled 'from scratch'. One effect of this approach is a potential loss of performance, caused by overheads such as additional synchronisation, loss of data locality, and reductions in the efficiency of loop unrolling and pipelining.

An attempt to ameliorate this loss of performance has led to the class of affinity scheduling algorithms (Markatos and LeBlanc [7], see also Subramanian and Eager [9], and Yan et al. [12] for variants of affinity scheduling). Rather than maintaining a single work queue these algorithms are based on per-processor work queues with exchange of work (chunks of the loop iteration) if required. The underlying assumption of affinity scheduling algorithms remains that each execution of a loop is independent of previous executions.

Recently, Bull et al. [2] and Bull [1] have proposed a substantially different loop scheduling algorithm, termed Feedback Guided Dynamic Loop Scheduling (FGDLS). The method studies the parallel loop

```
do sequential t = 1, nsteps
   do parallel i=1,n
      call LOOP_BODY(I)
   end do
end do
```

and considers that the workload is changing only slowly from one execution of the parallel loop to the next, so that observed timing information from the current execution of the loop can, and should, be used to guide the scheduling of the next execution of the same loop. In this way it is possible to limit the number of chunks into which the loop iterations are divided to be equal to the number of processors, and by careful design of the algorithm it is possible thereby to limit the loss of performance associated with guided self-scheduling algorithms and the synchronisation costs of affinity scheduling algorithms.

2 Feedback Guided Dynamic Loop Scheduling Algorithm

The problem to be considered is the scheduling, across p processors, of a loop that ranges from 1 to n. We assume that the scheduling algorithm has defined the following lower and upper loop iteration bounds (for the p processors) on time step (outer *sequential* iteration) t:

$$l_j^t, h_j^t \in N, \;\; j = 1, 2, \ldots, p,$$

where N are the natural numbers; note that

$$l_{j+1}^t = h_j^t + 1, \;\; j = 1, 2, \ldots, p-1,$$

and that

$$l_1^t = 1, \qquad h_p^t = n.$$

We further assume that the corresponding measured execution times on time step t are available and are given by $T_j^t, j = 1, 2, \ldots, p$. Given this execution time data, a piecewise constant approximation to the actual workload at time step t is given by

$$\hat{w}_i^t = \frac{T_j^t}{h_j^t - l_j^t + 1}, \;\; l_j^t \le i \le h_j^t, \;\; j = 1, 2, \ldots, p.$$

Note that an alternative view is that $\hat{w}_i^t$ is the mean observed workload per loop iteration index on time step t. The FGDLS algorithm defines new iteration bound limits for time step $t+1$, $l_j^{t+1}, h_j^{t+1} \in N, j = 1, 2, \ldots, p$, so that this piecewise constant function is approximately equipartitioned amongst the p processors:

$$\sum_{i=l_j^{t+1}}^{h_j^{t+1}} \hat{w}_i^t \approx \frac{1}{p} \sum_{i=1}^{n} \hat{w}_i^t = \frac{1}{p} \sum_{k=1}^{p} T_k^t, \;\; j = 1, 2, \ldots, p. \quad (1)$$

Note that $l_1^{t+1} = 1, h_p^{t+1} = n$, and $l_{i+1}^{t+1} = h_i^{t+1} + 1, i = 1, 2, \ldots, p-1$. Thus the observed workload at time step t is approximately equally distributed amongst the processors at time step $t+1$. Note that the observed workload would be equally distributed (exactly) amongst the processors, if we were to treat the iteration index i as a continuous (rather than discrete) variable and chose $\tilde{l}_j^{t+1}, \tilde{h}_j^{t+1} \in R, j = 1, 2, \ldots, p$, so that

$$\int_{\tilde{l}_j^{t+1}}^{\tilde{h}_j^{t+1}} \hat{w}^t(x)dx = \frac{1}{p} \int_1^n \hat{w}^t(x)dx = \frac{1}{p} \sum_{j=1}^{p} T_j^t, \quad (2)$$

where we extend the piecewise constant workload to a continuous domain by defining

$$\hat{w}^t(x) = \frac{T_j^t}{\tilde{h}_j^t - \tilde{l}_j^t}, \;\; x \in [\tilde{l}_j^t, \tilde{h}_j^t), \;\; j = 1, 2, \ldots, p. \quad (3)$$

Note that $\tilde{l}_1^{t+1} = 1, \tilde{h}_p^{t+1} = n$, and $\tilde{l}_{i+1}^{t+1} = \tilde{h}_i^{t+1}, i = 1, 2, \ldots, p-1$. But this approach results in non-integral values for the lower and upper loop iteration bounds. In fact the way that the FGDLS algorithm calculates the approximate iteration partition given in (1) is to solve the continuous problem (2) and then take $l_j^{t+1} = \left[\tilde{l}_j^{t+1}\right], h_j^{t+1} = l_j^{t+1} + 1, j = 1, 2, \ldots, p$, where $[x]$ denotes the integer closest to x.

The numerical performance of the FGDLS algorithm has been considered by Bull [1] for synthetic examples and by Bull et al. [2] for a real example (Numerical Weather Prediction). These papers demonstrate that the algorithm performs better than the alternatives when the workload does not vary too rapidly as the outer (sequential) iteration proceeds. However these papers do not address the question of convergence of the FGDLS algorithm, and the conditions on workload that guarantee convergence. Our aim in this paper is to derive conditions that guarantee convergence of the FGDLS algorithm.

3 Convergence

In this section we investigate the convergence properties of the FGDLS algorithm. In particular we investigate the fundamental property of convergence of the algorithm in the case of a fixed, static, workload. We can state this loop scheduling problem as:

3.1 Discrete Loop Scheduling Problem

The Discrete Loop Scheduling (DLS) Problem can be stated as follows:
Given a fixed workload $W_i^\star > 0, i = 1, 2, \ldots, N$, and $p < N$ processors, determine an assignment of work to the p processors so that the workload is equally distributed.

As stated the problem appears straightforward; the complication is that the workload $W_i^\star, i = 1, 2, \ldots, N$, is unknown and the FGDLS algorithm derives a schedule based on a “sample” of the unknown workload (at p points on each execution of the sequential outer iteration). The difficulty in considering the convergence of the algorithm in this discrete case is two-fold:

- the stated DLS problem may have no unique solution, or no solution at all (although there will exist “solution(s)” that balance the workload as well as possible - these results are true since there is only a finite number of possibilities to consider).

- the algorithm as stated in Section 2 proceeds via a continuous analogue and it is more appropriate to consider the convergence properties of that part of the algorithm.

Hence we consider instead the following continuous problem.

3.2 Continuous Loop Scheduling Problem

The Continuous Loop Scheduling (CLS) Problem can be stated as:

Given a fixed positive workload $W^\star(x) > 0, x \in [a,b], a < b$, *and* p *processors, determine an assignment of work to the processors so that the workload is equally distributed.*

In this case it is straightforward to show that there exists a unique solution to the problem: we choose $x_j^\star \in R,\ \ j = 1,2,\ldots,p-1,\ \ a = x_0^\star < x_1^\star < \cdots < x_p^\star = b$, so that

$$\int_{x_{j-1}^\star}^{x_j^\star} W^\star(x)dx = \frac{1}{p}\int_a^b W^\star(x)dx, \quad j = 1,2,\ldots,p. \tag{4}$$

Since $W^\star(x)$ is positive for $x \in [a,b]$, these equations uniquely define the optimal partition $\left\{x_0^\star, x_1^\star, \ldots, x_p^\star\right\}$ of $[a,b]$.

Suppose that we wish to use the continuous version of the FGDLS algorithm described by Equation (2) to generate a load balanced partition for the CLS problem. In this paper we show that this continuous FGDLS algorithm is convergent under fairly weak conditions on the workload $W^\star(x)$.

Let us suppose that the partition $\{x_j^t, j = 0,1,\ldots,p\}$ is known. Equation (3) defines the piecewise constant workload $\hat{w}^t(x)$ in terms of the measured execution times $T_j^t,\ \ j = 1,2,\ldots,p$, for this partition. (Note that, for this continuous problem,

$$T_j^t = \int_{x_{j-1}^t}^{x_j^t} W^\star(x)dx,$$

$j = 1,2,\ldots,p$.) We assume that, for the $t+1$ outer iteration, the algorithm defines the new partition $\{x_j^{t+1}, j = 0,1,\ldots,p\}$, where

$$\int_{x_j^{t+1}}^{x_{j+1}^{t+1}} \hat{w}^t(x)dx = \frac{1}{p}\int_a^b \hat{w}^t(x)dx, \ \ j = 0,\ldots,p-1. \tag{5}$$

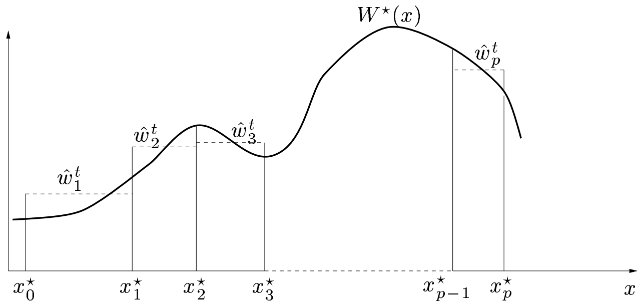

Figure 1. Piecewise Constant Workload

Note that $x_0^{t+1} = a$, $x_p^{t+1} = b$, $\forall\, t$.

Recall that the piecewise constant workload

$$\hat{w}^t(x) = \frac{1}{x_j^t - x_{j-1}^t}\int_{x_{j-1}^t}^{x_j^t} W^\star(x)dx, \ \ x \in \left[x_{j-1}^t, x_j^t\right)$$

is a constant function on each interval $\left[x_{j-1}^t, x_j^t\right)$.

Theorem 1 *The piecewise constant workload* $\hat{w}^t(x)$ *satisfies*

$$\int_a^{x_j^t} \hat{w}^t(x)dx = \int_a^{x_j^t} W^\star(x)dx \tag{6}$$

Proof

$$\int_a^{x_j^t} \hat{w}^t(x)dx = \sum_{i=1}^{j}\int_{x_{i-1}^t}^{x_i^t} \hat{w}^t(x)dx = \sum_{i=1}^{j}\int_{x_{i-1}^t}^{x_i^t} \hat{w}_i^t dx =$$

$$= \sum_{i=1}^{j}\left(x_i^t - x_{i-1}^t\right)\ \hat{w}_i^t = \sum_{i=1}^{j}\int_{x_{i-1}^t}^{x_i^t} W^\star(x)dx =$$

$$\int_a^{x_j^t} W^\star(x)dx.$$

Thus, the theorem is proved.

♠

An important consequence of Theorem 1 is that

$$\int_a^b \hat{w}^t(x)dx = \int_a^b W^\star(x)dx.$$

This implies that

$$\frac{1}{p}\int_a^b \hat{w}^t(x)dx := \overline{W}$$

is an invariant of FGDLS algorithm. Thus, the partitions $\{x_j^{t+1}, j = 0, 1, \ldots, p\}$ satisfy

$$\int_{x_j^{t+1}}^{x_{j+1}^{t+1}} \hat{w}^t(x)dx = \overline{W}, \;\; j = 0, 1, \ldots, p-1, \;\; \forall t. \qquad (7)$$

We now establish recursive equations for the sequence $\{x_j^t, j = 1, 2, \ldots, p-1\}$.

Theorem 2 *For* $x_j^t \leq x_j^*$,

$$x_j^{t+1} = x_{u(j)}^t + (x_{u(j)+1}^t - x_{u(j)}^t)\, \frac{\int_{x_{u(j)}^t}^{x_j^*} W^\star(x)dx}{\int_{x_{u(j)}^t}^{x_{u(j)+1}^t} W^\star(x)dx}, \quad (8)$$

where $u(j):\;\; x_{u(j)}^t \leq x_j^* < x_{u(j)+1}^t$.
For $x_j^t > x_j^*$,

$$x_j^{t+1} = x_{v(j)-1}^t + (x_{v(j)}^t - x_{v(j)-1}^t)\, \frac{\int_{x_{v(j)-1}^t}^{x_j^*} W^\star(x)dx}{\int_{x_{v(j)-1}^t}^{x_{v(j)}^t} W^\star(x)dx}. \quad (9)$$

where $v(j):\;\; x_{v(j)-1}^t < x_j^* < x_{v(j)}^t$.

Proof

Case 1. $x_j^t \leq x_j^*$ (see Figure 2).
Note that $j \leq u(j)$. ¿From Equation (7) we have

$$\int_a^{x_j^{t+1}} \widehat{w}^t(x)dx = j\;\overline{W} \Leftrightarrow$$

$$\int_a^{x_{u(j)}^t} \widehat{w}^t(x)dx + \int_{x_{u(j)}^t}^{x_j^{t+1}} \widehat{w}^t(x)dx = \int_a^{x_j^*} W^\star(x)dx.$$

Now, the first integral is reduced by Equation (6) and the second integral uses the fact that $\widehat{w}^t$ is constant on $[x_{u(j)}^t, x_j^{t+1})$, and that $x_j^{t+1} < x_{u(j)+1}^t$. Thus

$$\int_a^{x_{u(j)}^t} W^\star(x)dx + \int_{x_{u(j)}^t}^{x_j^{t+1}} \widehat{w}_{u(j)+1}^t dx = \int_a^{x_j^*} W^\star(x)dx$$

$$(x_j^{t+1} - x_{u(j)}^t)\;\widehat{w}_{u(j)+1}^t = \int_{x_{u(j)}^t}^{x_j^*} W^\star(x)dx$$

$$(x_j^{t+1} - x_{u(j)}^t)\;\frac{\int_{x_{u(j)}^t}^{x_{u(j)+1}^t} W^\star(x)dx}{x_{u(j)+1}^t - x_{u(j)}^t} = \int_{x_{u(j)}^t}^{x_j^*} W^\star(x)dx$$

$$x_j^{t+1} = x_{u(j)}^t + (x_{u(j)+1}^t - x_{u(j)}^t)\;\frac{\int_{x_{u(j)}^t}^{x_j^*} W^\star(x)dx}{\int_{x_{u(j)}^t}^{x_{u(j)+1}^t} W^\star(x)dx}.$$

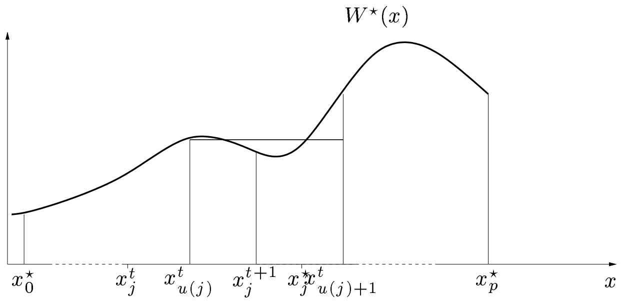

Figure 2. Recurrence of Partition Bounds; Case 1.

Case 2. $x_j^t > x_j^*$ (see Figure 3).
Note that $j \geq v(j)$. Using similar arguments to Case 1, from equation (7) we have

$$\int_a^{x_j^{t+1}} \widehat{w}^t(x)dx = j\;\overline{W}$$

$$\int_a^{x_{v(j)-1}^t} \widehat{w}^t(x)dx + \int_{x_{v(j)-1}^t}^{x_j^{t+1}} \widehat{w}^t(x)dx = \int_a^{x_j^*} W^\star(x)dx$$

$$\int_a^{x_{v(j)-1}^t} W^\star(x)dx + \int_{x_{v(j)-1}^t}^{x_j^{t+1}} \widehat{w}_{v(j)}^t dx = \int_a^{x_j^*} W^\star(x)dx$$

$$(x_j^{t+1} - x_{v(j)-1}^t)\;\frac{\int_{x_{v(j)-1}^t}^{x_{v(j)}^t} W^\star(x)dx}{x_{v(j)}^t - x_{v(j)-1}^t} = \int_{x_{v(j)-1}^t}^{x_j^*} W^\star(x)dx$$

$$x_j^{t+1} = x_{v(j)-1}^t + (x_{v(j)}^t - x_{v(j)-1}^t)\;\frac{\int_{x_{v(j)-1}^t}^{x_j^*} W^\star(x)dx}{\int_{x_{v(j)-1}^t}^{x_{v(j)}^t} W^\star(x)dx}.$$

♠

Thus, Equations (8,9) give the recurrences for the partition sequences. Based on these we can obtain some monotonic properties for the sequence $\{x_j^t\}_{t>0}$. For Case 1, Equation (8) directly shows that $x_j^{t+1} \geq x_{u(j)}^t \geq x_j^t$. Moreover, since

$$\int_a^{x_j^{t+1}} \widehat{w}^t(x)dx = \int_a^{x_j^*} W^*(x)dx <$$

$$< \int_a^{x_{u(j)+1}^t} W^*(x)dx = \int_a^{x_{u(j)+1}^t} \widehat{w}^t(x)dx$$

we also find that that $x_j^{t+1} < x_{u(j)+1}^t$. For Case 2, the inequalities $x_j^{t+1} \geq x_{v(j)-1}^t$ and $x_j^{t+1} < x_j^t$ are similarly

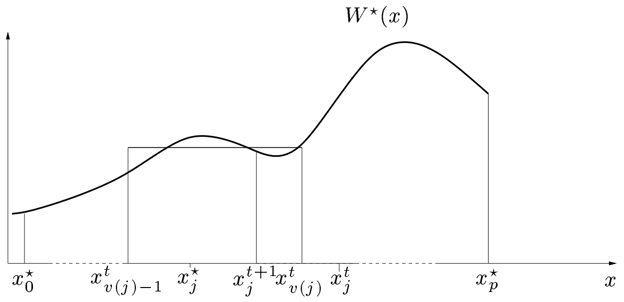

Figure 3. Recurrence of Partition Bounds; Case 2.

deduced from Equation (9). Thus, the sequence $\{x_j^t\}_{t>0}$ satisfies

$$x_j^t \le x_j^* \Rightarrow x_j^t \le x_j^{t+1} \tag{10}$$

$$x_j^t > x_j^* \Rightarrow x_j^t > x_j^{t+1} \tag{11}$$

In the following section, Equations (8,9) are used to study the convergence of the partition sequence $\{x_j^t\}_{t>0}$.

4 Convergence of FGDLS

In this section, we prove that the bound sequences are convergent when the workload $W^\star$ satisfies the following equation

$$\sup_{u\in[a,b]} W^\star(u) < 2 \inf_{u\in[a,b]} W^\star(u). \tag{12}$$

Theorem 3 *Assuming that $W^\star(x), x \in [a,b]$ is continuous then the partition sequence $\{x_j^t\}_{t>0}$ satisfies the following inequalities:*

$$\left|x_j^{t+1} - x_j^*\right| \le \sup_{v,u\in[a,b]} \left|1 - \frac{W^\star(u)}{W^\star(v)}\right| \; \left|x_j^t - x_j^*\right|. \tag{13}$$

Proof Let the index $j \in [1, p-1]$. Recall that the sequence $\{x_j^t\}_{t>0}$ satisfies Equations (8,9). Two cases are analysed as follows.

Case 1. $x_j^t \le x^t_{u(j)} \le x_j^* < x^t_{u(j)+1}$
Equation (8) can be rewritten as:

$$(x_j^{t+1} - x^t_{u(j)}) \frac{\int_{x^t_{u(j)}}^{x^t_{u(j)+1}} W^\star(x)dx}{(x^t_{u(j)+1} - x^t_{u(j)})} =$$

$$= (x_j^* - x^t_{u(j)}) \frac{\int_{x^t_{u(j)}}^{x_j^*} W^\star(x)dx}{(x_j^* - x^t_{u(j)})}.$$

Given that $W^\star(x), x \in [a,b]$ is continuous, by the mean value theorem for integrals, $\exists c_1 \in (x^t_{u(j)}, x^t_{u(j)+1})$ such that

$$\frac{\int_{x^t_{u(j)}}^{x^t_{u(j)+1}} W^\star(x)dx}{(x^t_{u(j)+1} - x^t_{u(j)})} = W^\star(c_1)$$

and $\exists c_2 \in (x^t_{u(j)}, x_j^*)$ such that

$$\frac{\int_{x^t_{u(j)}}^{x_j^*} W^\star(x)dx}{(x_j^* - x^t_{u(j)})} = W^\star(c_2).$$

Thus we have

$$(x_j^{t+1} - x^t_{u(j)}) W^\star(c_1) = (x_j^* - x^t_{u(j)}) W^\star(c_2)$$

$$(x_j^{t+1} - x_j^*) W^\star(c_1) = (x_j^* - x^t_{u(j)})(W^\star(c_2) - W^\star(c_1))$$

$$x_j^{t+1} - x_j^* = (x_j^* - x^t_{u(j)}) \frac{(W^\star(c_2) - W^\star(c_1))}{W^\star(c_1)}.$$

Hence

$$\left|x_j^{t+1} - x_j^*\right| = \left|x_j^* - x^t_{u(j)}\right| \left|\frac{W^\star(c_2) - W^\star(c_1)}{W^\star(c_1)}\right| =$$

$$= \left|1 - \frac{W^\star(c_2)}{W^\star(c_1)}\right| \cdot \left|x^t_{u(j)} - x_j^*\right| \le$$

$$\le \sup_{v,u\in[a,b]} \left|1 - \frac{W^\star(u)}{W^\star(v)}\right| \left|x^t_{u(j)} - x_j^*\right|.$$

Now, let us remark that, since $x_j^t \le x^t_{u(j)} \le x_j^*$, we have that $\left|x^t_{u(j)} - x_j^*\right| \le \left|x_j^t - x_j^*\right|$. Therefore Equation (13) holds.

Case 2. $x_j^t \ge x^t_{v(j)} > x_j^* \ge x^t_{v(j)-1}$
Firstly, we transform Equation (9) as follows:

$$x_j^{t+1} = x^t_{v(j)-1} + (x^t_{v(j)} - x^t_{v(j)-1}) \frac{\int_{x^t_{v(j)-1}}^{x_j^*} W^\star(x)dx}{\int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx}$$

$$\frac{(x_j^{t+1} - x^t_{v(j)-1})}{(x^t_{v(j)} - x^t_{v(j)-1})} \int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx = \int_{x^t_{v(j)-1}}^{x_j^*} W^\star(x)dx$$

$$\int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx - \frac{x_j^{t+1} - x^t_{v(j)-1}}{x^t_{v(j)} - x^t_{v(j)-1}} \int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx =$$

$$= \int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx - \int_{x^t_{v(j)-1}}^{x_j^*} W^\star(x)dx$$

$$\left(1 - \frac{x_j^{t+1} - x^t_{v(j)-1}}{x^t_{v(j)} - x^t_{v(j)-1}}\right) \int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx =$$

$$= \int_{x_j^*}^{x^t_{v(j)}} W^\star(x)dx$$

which it finally gives

$$(x^t_{v(j)} - x^{t+1}_j)\frac{\int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx}{(x^t_{v(j)} - x^t_{v(j)-1})} =$$

$$= (x^t_{v(j)} - x^*_j)\frac{\int_{x^*_j}^{x^t_{v(j)}} W^\star(x)dx}{(x^t_{v(j)} - x^*_j)}.$$

Now, by the mean value theorem, $\exists c_3 \in (x^t_{v(j)-1}, x^t_{v(j)})$ such that

$$\frac{\int_{x^t_{v(j)-1}}^{x^t_{v(j)}} W^\star(x)dx}{(x^t_{v(j)} - x^t_{v(j)-1})} = W^\star(c_3),$$

and $\exists c_4 \in (x^*_j, x^t_{v(j)})$ such that

$$\frac{\int_{x^*_j}^{x^t_{v(j)}} W^\star(x)dx}{(x^t_{v(j)} - x^*_j)} = W^\star(c_4).$$

Thus

$$(x^t_{v(j)} - x^{t+1}_j)W^\star(c_3) = (x^*_j - x^t_{v(j)})W^\star(c_4).$$

Following identical arguments to Case 1, we find that

$$\left|x^{t+1}_j - x^*_j\right| \leq \sup_{v,u\in[a,b]}\left|1 - \frac{W^\star(u)}{W^\star(v)}\right|\left|x^t_{v(j)} - x^*_j\right|.$$

Since $x^t_j \geq x^t_{v(j)} > x^*_j$, we find that $\left|x^t_{v(j)} - x^*_j\right| \leq \left|x^t_j - x^*_j\right|$. Therefore, Equation (13) is true.

Thus, the theorem is proved.

♠

Since, $W^\star : [a,b] \to (0,\infty)$ is a continuous function, the function $W^\star_1 : [a,b] \times [a,b] \to (0,\infty)$ defined by $W^\star_1(u,v) = \left|1 - \frac{W^\star(u)}{W^\star(v)}\right|$ is continuous on the compact set $[a,b]\times[a,b]$. The supremum $\sup_{(u,v)\in[a,b]\times[a,b]} W^\star_1(u,v)$ exists because the function $W^\star_1$ achieves its upper bound on the compact set.

Theorem 4 *Given that $W^\star : [a,b] \to (0,\infty)$ is a continuous function, then*

$$\sup_{u,v\in[a,b]}\left|1 - \frac{W^\star(u)}{W^\star(v)}\right| < 1 \Leftrightarrow$$

$$\sup_{u\in[a,b]} W^\star(u) < 2\inf_{u\in[a,b]} W^\star(u).$$

Proof Now,

$$\sup_{v,u\in[a,b]}\left|1 - \frac{W^\star(u)}{W^\star(v)}\right| < 1 \Leftrightarrow$$

$$\left|1 - \frac{W^\star(u)}{W^\star(v)}\right| < 1, \quad \forall u,v \in [a,b] \quad \Leftrightarrow$$

$$-1 < 1 - \frac{W^\star(u)}{W^\star(v)} < 1, \quad \forall u,v \in [a,b] \quad \Leftrightarrow$$

$$0 < \frac{W^\star(u)}{W^\star(v)} < 2, \quad \forall u,v \in [a,b] \quad \Leftrightarrow$$

$$W^\star(u) < 2W^\star(v), \quad \forall u,v \in [a,b] \quad \Leftrightarrow$$

$$\sup_{u\in[a,b]} W^\star(u) < 2\inf_{u\in[a,b]} W^\star(u).$$

♠

Theorem 5 *Provided that*

$$\sup_{u\in[a,b]} W^\star(u) < 2\inf_{u\in[a,b]} W^\star(u),$$

then the sequences generated by the FGDLS algorithm converge to the solution of the CLS problem.

Proof By Theorem 4,

$$\sup_{u\in[a,b]} W^\star(u) < 2\inf_{u\in[a,b]} W^\star(u) \Leftrightarrow$$

$$q := \sup_{v,u\in[a,b]}\left|1 - \frac{W^\star(u)}{W^\star(v)}\right| < 1.$$

Then by Theorem 3,

$$\left|x^{t+1}_j - x^*_j\right| \leq q\left|x^t_j - x^*_j\right|, \quad j = 1,2,\ldots,p-1.$$

By repeatedly applying the result, we find that

$$\left|x^t_j - x^*_j\right| \leq q^{t-1}\left|x^1_j - x^*_j\right|, \quad j = 1,2,\ldots,p-1.$$

Since $q < 1$ it follows that the sequence $\left\{x^t_j\right\}_{t>0}$ is convergent and $\lim_{t\to\infty} x^t_j = x^*_j$.

♠

Equation (12) provides the simplest condition for the convergence of the bound sequences. It states that, over the range of interest for the iteration index, the maximum of the workload should be less than two times the minimum of the workload; this is equivalent to requiring that the workload does not have large variations across the range of the iteration index.

Example

The initial workload $W^* : [0,10] \to (0,\infty)$ is defined by $W^*(x) = 2 + \frac{1}{2}\cdot\sin\frac{2\pi x}{10}$. Remark that $\sup_{u\in[0,10]} W^*(u) = \frac{5}{2} < 2\cdot\inf_{u\in[0,10]} W^*(u) = 3$, therefore Equation (12) holds. The initial partition is given by $x^1 = (0,\ 2.5,\ 5,\ 7.5,\ 10)$. The partitions for the first 6 iterations are presented below.

t=1	0	2.5	5	7.5	10
t=2	0	2.156	3.406	5.539	10
t=3	0	2.166	3.738	6.069	10
t=4	0	2.172	3.939	6.352	10
t=5	0	2.176	4.059	6.530	10
t=6	0	2.179	4.132	6.649	10

The execution times for these partitions are as follows.

t=1	5.795	5.795	4.204	4.204
t=2	4.938	3.099	4.586	7.374
t=3	4.962	3.867	4.726	6.443
t=4	4.979	4.320	4.726	5.973
t=5	4.989	4.587	4.735	5.688
t=6	4.994	4.747	4.758	5.499

Moreover, the results for $t = 100$ are:
- the partition $x^{100} = (0,\ 2.181,\ 4.247,\ 6.972,\ 10)$
- the execution times $T^{100} = (5.0,\ 5.0,\ 5.0,\ 5.0)$
which represent the optimal solution.

Now, we consider the workload $W^* : [0, 10] \to (0, \infty)$ defined by $W^*(x) = 200 + 100 \cdot \sin\frac{2\pi x}{10}$. Since,$\sup_{u\in[0,10]} W^*(u) = 300 > 2 \cdot \inf_{u\in[0,10]} W^*(u) = 200$, Equation (12) does not hold. For this workload we have found that the partition sequences are periodic and thus they are not convergent.

5 Conclusions

In the earlier papers by Bull [1] and Bull et al. [2] it has been observed that the FGDLS algorithm performs particularly well for problems where the workload does not vary too rapidly as the outer (sequential) iteration proceeds. However, the question of convergence of the FGDLS algorithm has not previously been investigated. In this paper we have presented a proof for the *global* convergence of the FGDLS for the Continuous Loop Scheduling (CLS) Problem. The major condition of the theorem is that the workload is continuous and satisfies $\sup\limits_{u\in[a,b]} W^\star(u) < 2 \inf\limits_{u\in[a,b]} W^\star(u)$. (This condition essentially requires that the workload does not vary too rapidly across the index interval $[a, b]$.)

6 Acknowledgements

The second author acknowledges the support of the EPSRC research grant GR/K85704, “Feedback-guided Affinity Loop Scheduling for Multiprocessors”.

References

[1] Bull, J.M. (1998) *Feedback Guided Loop Scheduling: Algorithms and Experiments*, Proceedings of Euro-Par’98, Lecture Notes in Computer Science, Springer-Verlag.

[2] Bull, J.M., Ford, R.W. and Dickinson, A. (1996) *A Feedback Based Load Balance Algorithm for Physics Routines in NWP*, Proceedings of Seventh ECMWF Workshop on the Use of Parallel Processors in Meteorology, World Scientific.

[3] Eager, D.L. and Zahorjan, J. (1992) *Adaptive Guided Self-Scheduling*, Technical Report 92-01-01, Department of Computer Science and Engineering, University of Washington.

[4] Hummel, S.F., Schonberg, E. and Flynn, L.E. (1992) *Factoring: A Practical and Robust Method for Scheduling Parallel Loops*, Communications of the ACM, vol. 35, no. 8, pp. 90–101.

[5] Kruskal, C.P. and Weiss, A. (1985) *Allocating Independent Subtasks on Parallel Processors*, IEEE Trans. on Software Engineering, vol. 11, no. 10, pp. 1001–1016.

[6] Lucco, S. (1992) *A Dynamic Scheduling Method for Irregular Parallel Programs*, in Proceedings of ACM SIGPLAN ’92 Conference on Programming Language Design and Implementation, San Francisco, CA, June 1992, pp. 200–211.

[7] Markatos, E.P. and LeBlanc, T.J. (1994) *Using Processor Affinity in in Loop Scheduling on Shared Memory Multiprocessors*, IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 4, pp. 379–400.

[8] Polychronopoulos, C.D. and Kuck, D.J. (1987) *Guided Self-Scheduling: A Practical Scheduling Scheme for Parallel Supercomputers*, IEEE Transactions on Computers, vol. C-36, no. 12, pp. 1425–1439.

[9] Subramanian, S. and Eager, D.L. (1994) *Affinity Scheduling of Unbalanced Workloads*, in Proceedings of Supercomputing’94, IEEE Comp. Soc. Press, pp. 214–226.

[10] Tang, P. and Tew, P-C. (1986) *Processor Self-Scheduling for Multiple Nested Parallel Loops*, in Proceedings of 1986 Int. Conf. on Parallel Processing, pp. 528–535, St. Charles, IL.

[11] Tzen, T.H. and Ni, L.M. (1993) *Trapezoid Self-Scheduling Scheme for Parallel Computers*, IEEE Trans. on Parallel and Distributed Systems, vol. 4, no. 1, pp. 87–98.

[12] Yan, Y., Jin, C. and Zhang, X. (1997) *Adaptively Scheduling Parallel Loops in Distributed Shared-Memory Systems*, IEEE Trans. on Parallel and Distributed Systems, vol. 8, no. 1, pp. 70–81.