GENERALIZED ASYNCHRONOUS ITERATIONS

Aydin Uresin and Michel Dubois

Computer Research Institute
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA90077 USA

ABSTRACT

Asynchronous iterative methods for multiprocessors are generalized to relaxation techniques involving discrete variables. Asynchronous algorithms are more efficient than synchronized algorithms in multiprocessors because processes do not have to wait on each other at synchronization points. Sufficient conditions for the convergence of generalized asynchronous iterations are given and proved. Applications of the theory presented in this paper include asynchronous relaxation algorithms for scene labeling in image processing applications.

1. INTRODUCTION

Asynchronous iterative methods are very attractive for multiprocessors because they remove the need for synchronization between the processors cooperating in the algorithm. It has been shown [ROB79, DUB82] that synchronization between processes is a major source of performance degradation in multiprocessor algorithms. The reason is that the processes that synchronize have to wait for the *slowest* among them [KUN76]. Randomness in execution time of the different processes between two successive synchronization points is caused by the difference in the computation time of each process because they perform different functions or because they perform the same function on different input data sets, and by conflicts for shared resources such as an interconnection link or a shared-memory module. When the iteration is asynchronous, the processors never wait on each other and they can compute at full speed.

Asynchronous iterative methods have been applied so far to the solution of systems of equations. The variables manipulated in the chaotic relaxation scheme of Chazan and Miranker [CHA69] and in the purely asynchronous algorithm described by Baudet [BAU78] are continuous and defined in the vectorial space R^n. The definition that is given in Baudet's paper of a contracting operator is based on a metric defined in R^n. Since all his proofs are based on the definition of contracting operators, they cannot be applied as such to relaxation techniques in discrete fields.

Many relaxation techniques involve the manipulation of symbolic or discrete-valued data. One major example is scene labeling relaxation used in image understanding applications to define features of a digital pictures [ROS76]. In the general formulation of the problem, a set of labels is associated with objects (objects may be pixels in a digitized picture); labels are related to objects by compatibility relations. Certain labels may be incompatible with some objects; also for each pair of objects some pair of labels may not be compatible. The scene labeling algorithm finds the greatest consistent labeling. "Consistent" means that the compatibility relations are respected and "greatest" means that the maximum number of compatible labels has been associated with each object. In this paper, we also consider the case of fuzzy labeling.

In the following, we first define the generalization of asynchronous iterative algorithms in which the relaxed variables belong to an arbitrary set S. In this context, the generalized definition of a contracting operator is introduced. Then the convergence is proved for the general case. The theory is applied to the scene labeling problem, including fuzzy labeling.

2. CONVERGENCE OF ASYNCHRONOUS ITERATIONS

The following definition of asynchronous iterations is similar to the one given by Baudet [BAU78]. In Baudet's paper, only operators from R^n to R^n are considered. The following is the generalization of asynchronous iterations for the case of an operator F from S^n to S^n, where S is any set, finite or infinite, countable or not.

Definition 2.1: Generalized Asynchronous Iteration

Let F be an operator from S^n to S^n. An asynchronous iteration corresponding to the operator F and starting with a given vector $x(0)$ is a sequence $x(j)$, $j = 0, 1,...$ of vectors of S^n and defined recursively by :

$$x_i(j) = \begin{cases} x_i(j-1) & \text{if } i \notin J_j \\ F_i(x_1(s_1(j)),..., x_n(s_n(j)) & \text{if } i \in J_j, \end{cases} \qquad (1)$$

where $J = \{ J_j \mid j = 1,2,...\}$ is a sequence of nonempty subsets of $\{1,2,...,n\}$,

$C = \{s_1(j),..., s_n(j)) \mid j = 1,2,...\}$ is a sequence of elements in N^n and S is a set.

In addition, J and C are subject to the following conditions, for each i =1,...,n:

(a) $s_i(j) \leq j-1$, j = 1,2,...;

(b) $s_i(j)$, considered as a function of j, tends to infinity as j tends to infinity;

(c) i occurs infinitely many often in the sets J_j, j =1,2,...

An asynchronous iteration corresponding to F, starting with $x(0)$, and defined by J and C, will be denoted by $(F, x(0), J, C)$.

The above definition is a straightforward generalization of Baudet's definition for asynchronous iterative algorithms. An asynchronous algorithm is easily obtained by deriving first a synchronized multitasked algorithm, and by removing the synchronizations. Critical sections may still be necessary in order to preserve the integrity of shared data. In Baudet's paper, the significance and implications of definition 2.1 are further discussed in the context of numerical problems.

Definition 2.2 : Generalized Contracting Operator

An operator F from S^n to S^n is a generalized contracting operator on a subset

$$D = D_1 \text{x} D_2 \text{x...x} D_n \text{ of } S^n \text{ iff } \lim_{k \to \infty} F^k(D) = \{ \varsigma \}, \qquad (2)$$

where ς is the point of convergence, and $F(D)$ is defined as

$$F(D) = \{F_1(x) \mid x \in D\} \text{x} \{F_2(x) \mid x \in D\} \text{x...x} \{F_n(x) \mid x \in D\}, \qquad (3)$$

and F^k is the set obtained by applying this operator k times on D.

The above definition is a generalization of the two conditions given by Baudet for the convergence of an asynchronous iteration on R^n in the case where the domain D is a rectangle in n dimension. These conditions are that the operator F is contracting and that F(D) $\subseteq$ D. It is clear that in the case of iterations defined in R^n, these conditions are equivalent to (2). It is also clear that condition (2) is a sufficient condition for convergence of the synchronous algorithm defined on any set. The following two lemmas and theorem demonstrate that the condition is also sufficient for the asynchronous counterpart, and for any sequences J and C satisfying the conditions of definition 2.1.

Lemma 2.1 : If an operator F from S^n to S^n is a generalized contracting operator on a subset $D = D_1 x D_2 x ... x D_n$ of S^n, then

$$\{ \varsigma \} \subseteq F^{p+1}(D) \subseteq F^p(D), \text{for } all\ p = 0,1,2,... \tag{4}$$

Proof :

Definition 2.2 implies that for each subset E of D such that $E = E_1 x\ E_2 x ... x\ E_n$ and $\{ \varsigma \}$ $\subset$ E, there is an integer n which satisfies:

$$\{ \varsigma \} \subseteq F^k(D) \subset \text{E, for all } k \geq n \tag{5}$$

Now, suppose that there is an integer p such that $F^p(D) \subset F^{p+1}(D)$. Then,

$$F^{p+1}(D) = F^p(D) \cup \text{R}$$

$$\rightarrow \quad \text{F}\ \{F^{p+1}(D)\} = \text{F}\ \{F^p(D) \cup \text{R}\}$$

$$\rightarrow \quad F^{p+2}(D) \subseteq F^{p+1}(D)$$

$$\rightarrow \quad F^p(D) \subset F^{p+2}(D)$$

Similarly, by mathematical induction,

$$F^p(D) \subset F^q(D),\ q > p. \tag{6}$$

If we choose E = D $\cap$ $F^p(D)$, we cannot find an integer n such that

$$F^k(D) \subset \text{E, for all } k \geq n,$$

which leads to a contradiction and proves

$$F^{p+1}(D) \subseteq F^p(D),\ p = 0,1,2,....$$

On the other hand, (5) implies that

$$\{ \varsigma \} \subseteq F^p(D),\ p = 0,1,2,...$$

Therefore, we obtain

$$\{ \varsigma \} \subseteq F^{p+1}(D) \subseteq F^p(D),\ p = 0,1,2,...$$

Lemma 2.2 : If F is a generalized contracting operator from S^n to S^n on a subset $D = D_1 x D_2 x ... x D_n$ of S^n, then

$$F_i{}^{p+1}(D) \subseteq F_i{}^p(D),\ i = 1,2,...,n, \tag{7}$$

where $F_i{}^p(D)$ is defined by

$$F^p(D) = F_1{}^p(D) x F_2{}^p(D) x ... x F_n{}^p(D) \tag{8}$$

Proof:

Let $E = F^p(D)$. Since $E = E_1 x E_2 x ... x E_n$, we have $F(E) = F_1(E) x F_2(E) x ... x F_n(E)$, and, from lemma 2.1, $F(E) \subseteq E$.

The claim of the lemma follows.

The following theorem proves the convergence of generalized asynchronous iterations when the operator is a generalized contracting operator.

Theorem 2.1 : If F is a generalized contracting operator on a subset $D = D_1 x D_2 x ... x D_n$ of S^n, then an asynchronous iteration $(F, x(0), J, C)$ corresponding to F and starting with a vector $x(0)$ in D converges to a unique fixed point of F in D.

Proof:

We will show that for any $p \in \{0,1,2,...\}$, an integer j_p can be obtained such that the sequence of iterates of $(F, x(0), J, C)$ satisfies

$$x(j) \in F^p(D), \text{ for } j \geq j_p \tag{9}$$

We first show that (9) holds for $p = 0$. If we let $j_0 = 0$, then for $j \geq 0$, we have:

$$x(j) \in D \tag{10}$$

(10) is true for $j = 0$, since $x(0)$ is in D. Assume that it is true for $0 \leq j < k$ and consider $x(k)$. Let z denote the vector with components $z_i = x_i(s_i(k))$, for $i = 1,2,...,n$. From definition 2.1, the components of $x(k)$ are given either by $x_i(k) = x_i(k-1)$ if $i \notin J_k$, in which case $x_i(k) = x_i(k-1) \in D_i$, or by $x_i(k) = F_i(z)$ if $i \in J_k$. In this latter case, we note that, as $s_i(k) < k$ and $z \in D$, we have $F(z) \in D$.

This result in turn implies that $x_i(k) = F_i(z) \in D_i$, and that $x(k) \in D$. (10) is proved by induction, which shows that (9) is true for $p = 0$ if we choose $j_0 = 0$. Assume now that a j_p has been found to satisfy (9) for $0 \leq p < q$. First, define r by :

$$r = \text{Min } \{ k \mid \text{for all } j \geq k, s_i(j) \geq j_{q-1}, i=1,...,n \}$$

We see from condition (b) of definition 2.1 that this number exists, and we note that, from condition (a), we have $r > j_{q-1}$ which shows in particular that $x(r) \in F^{q-1}(D)$.

Then, take $j \geq r$ and consider the components of $x(j)$. As above, let z be the vector with components $z_i = x_i(s_i(j))$. From the choice of r, we have $s_i(j) \geq j_{q-1}$, for $i = 1,...,n$ and this shows that $z \in F^{q-1}(D)$, and $F(z) \in F^q(D)$. This shows that, if $i \in J_j$, $F_i(z)$ satisfies $F_i(z) \in F_i{}^q(D)$, $i = 1,2,...n$, and we obtain

$$x_i(j) \in F_i{}^q(D), \; i = 1,2,...,n \tag{11}$$

This result means that as soon as the ith component is updated between the rth and the jth iteration we have (11). On the other hand, if $i \notin J_j$, the ith component is not modified.

Now, define j_q as:

$$j_q = Min \{ j \mid j \geq r, \text{ and } \{1,...,n\} = J_r \cup ... \cup J_j \}$$

This number exists by condition (c) of definition 2.1, and for any $j \geq j_q$ every component is updated at least once between the rth and the jth iterations and therefore (11) holds for $i = 1,...,n$. This shows that (9) holds for $p = q$, and by induction (9) holds for

$p = 0,1,\ldots$ Since p can be chosen arbitrarily large, and therefore $F^p(\mathrm{D})$ can be made arbitrarily small, we obtain

$$\lim_{j \to \infty} x(j) = \varsigma,$$

which is the desired result.

The following section illustrates the application of theorem 2.1 to the problems of the discrete and fuzzy scene labeling described in [ROS76].

3. SCENE LABELING (DISCRETE MODEL)

The following definitions are drawn from [ROS76].

Let $\mathrm{A} = \{a_1,\ldots,a_n\}$ be the set of objects to be labeled and $\Lambda = \{\lambda_1,\ldots,\lambda_n\}$, the set of possible labels. For any given object a_i, not every label in Λ may be appropriate. Let Λ_i, be the set of labels which are compatible with object a_i, i = 1,...,n. For each pair of objects (a_i, a_j) some labels may be compatible, while others are not.

Let $\Lambda_{ij} \subseteq \Lambda_i \times \Lambda_j$ be the set of compatible pairs of labels; thus $(\lambda, \lambda') \in \Lambda_{ij}$ means that it is possible to label a_i with label λ and a_j with label λ'. If a_i and a_j are irrelevant to one another, then there are no restrictions on the possible pairs of labels that they can have, so that $\Lambda_{ij} = \Lambda_i \times \Lambda_j$.

By a labeling $\mathrm{L} = (L_1,\ldots,L_n)$ of A, we mean an assignement of a set of labels $L_i \subseteq \Lambda$ to each $a_i \in \mathrm{A}$. The labeling is consistent if for all i,j we have $(\{\lambda\} \times \Lambda_j) \cap \Lambda_{ij} \neq \phi$, for all $\lambda \in L_i$.

We say that a labeling $\mathrm{L} = \{L_1,\ldots,L_n\}$ contains another labeling L' if $L_i' \subseteq L_i$ for i=1,...,n. The greatest labeling L^∞ is a consistent labeling such that any other consistent labeling is contained in L^∞.

According to this model, the discrete relaxation procedure operates as follows. It starts with the initial labeling $\mathrm{L}(0) = \{\Lambda_1,\ldots,\Lambda_n\}$. During each step, we eliminate from each L_i all labels λ, such that $(\{\lambda\} \times L_j) \cap \Lambda_{ij} = \phi$ for some j. Thus we discard a label λ from object a_i if there exists an object a_j such that no label compatible with λ is assigned to (a_i, a_j). If for all j's, such that j = 1,...,n, and j $\neq$ i there exists a label $\lambda' \in L_j$ and λ' is compatible with λ, then we keep the label λ in L_i. We shall refer to the operation executed at each iteration as Δ.

Lemma 3.1 : [ROS76]

$$L^\infty \subseteq \ldots \subseteq \mathrm{L}(k) \subseteq \ldots \subseteq \mathrm{L}(0). \tag{12}$$

Theorem 3.1 : [ROS76]

$$\lim_{k \to \infty} L(k) = L^\infty \tag{13}$$

The following lemma and theorem prove the convergence of the discrete labelling relaxation implemented as an asynchronous iteration.

Lemma 3.2 : If L and L' are labelings such that $\mathrm{L} \subseteq \mathrm{L'}$ then $\Delta(\mathrm{L}) \subseteq \Delta(\mathrm{L'})$

Proof:

Let E = L - Δ (L), E' = L - Δ (L') and suppose that Δ (L') $\subset$ Δ (L). Then, from Lemma 3.1, Δ (L) $\subseteq$ L and therefore E $\subset$ E', which implies that there exists a $\lambda \in L_i$ such that

$$(\{\lambda\} \times L_j') \cap \Lambda_{ij} = \phi, \text{ and}$$

$$(\{\lambda\} \times L_j) \cap \Lambda_{ij} = \phi,$$

for some i,j pair. However, this is not possible, since $L_j \subseteq L_j'$. This contradiction completes the proof.

Theorem 3.2 : An asynchronous iteration (Δ, L(0), J, C) converges to L^{∞}.

Proof :

Let D(k) = { L | $L^{\infty} \subseteq$ L $\subseteq$ L(k) }. From Lemma 3.2, Δ (D(k)) $\subseteq$ D(k+1).

Assume Δ^{k} (D(0)) $\subseteq$ D(k) (14)

Then, Δ^{k+1} (D(0)) $\subseteq$ D(k+1)

since (14) is true for k=0, and therefore, by mathematical induction, it is also true for k=1,2,...

On the other hand, from theorem 3.1, $\lim_{k \to \infty} L(k) = L^{\infty}$.

As a result, $\lim_{k \to \infty} D(k) = \{L^{\infty}\}$ (15)

Therefore, $\lim_{k \to \infty} \Delta^{k}$ (D(0)) = $\{L^{\infty}\}$ (16)

which proves that Δ is a contracting operator on D(0). Since the initial labeling, L(0), is in D(0), from Theorem 2.1, the sequence L(k) converges to L^{∞}.

In an asynchronous multiprocessor implementation of the scene labeling algorithm, each processor is assigned a subset of the objects to classify. The relaxation process does not require any synchronization. A process can freely access the set of label currently associated with objects processed by different processors. Critical sections may be needed to access the label set.

4. SCENE LABELING (FUZZY MODEL)

In this model, A and Λ are defined as in the discrete case, and for each i we are given a fuzzy label set Λ_i associated with the object a_i. This Λ_i is a fuzzy subset of Λ, i.e., a mapping from Λ into the interval [0,1]. In addition, for each pair of objects (a_i, a_j), where i $\neq$ j, we are given a fuzzy set Λ_{ij} of pairs of labels; this is a mapping from $\Lambda \times \Lambda$ into [0,1]. Here, we assume that

$$\Lambda_{ij}(\lambda, \lambda') \leq \inf(\Lambda_i(\lambda), \Lambda_j(\lambda'))$$

for all i,j,λ,λ'. By a fuzzy labeling L = (L_1,...,L_n) of A we mean an assignement of a fuzzy subset L_i of Λ to each a_i, i=1,...,n. We say that $L \leq L'$ if $L_i \leq L_i'$, i=1,...,n (i.e., $L_i(\lambda) \leq L_i'(\lambda')$). We also define sup (L, L') = (sup (L_1,L_1'), ..., sup (L_n ,L_n')). The fuzzy labeling L is called consistent if, for all i,j, and λ, we have

$$\sup\ [\inf(L_j(\lambda'), \Lambda_j(\lambda, \lambda'))] \geq L_i(\lambda) \text{ over all } \lambda'$$

The label weakening algorithm is defined as follows. We start with the initial fuzzy labeling L(0) = (Λ_1, ..., Λ_n). At each iteration, we apply the operation Ω, defined as follows:

$$\Omega_i(L(\lambda)) = \inf \{ \sup [\inf (L_j(\lambda'), \Lambda_{ij}(\lambda,\lambda'))] \} \text{ over all } \lambda' \text{ and } j \tag{17}$$

Lemma 4.1 [ROS76]

$$L^{\infty} \leq \cdots \leq L(k+1) \leq L(k) \leq \cdots < L(0), \tag{18}$$

where L^{∞} is the greatest consistent labeling.

Theorem 4.1 [ROS76]

$$\lim_{k \to \infty} L(k) = L^{\infty} \tag{19}$$

Lemma 4.1

If L and L' are fuzzy labelings such that $L \leq L'$ then Ω (L)$\leq$ Ω (L').

Proof:

The proof is obvious from (17)

Theorem 4.2

An asynchronous iteration (Ω, L(0), J, C) converges to L^{∞}.

Proof:

If we substitute $\subseteq$ with $\leq$ and Δ with Ω, the proof of theorem 4.2 is exactly the same as the proof of theorem 3.2

5. CONCLUSION

In this paper, we have introduced a sufficient condition for the convergence of asynchronous parallel relaxation of discrete data in multiprocessors systems. To illustrate the algorithm, we have applied the main theorem (theorem 2.1) to the cases of discrete and fuzzy labelings of a set of objects. These algorithms are important algorithms in image processing and understanding. The results presented here are extensions of results published in [BAU78] and [ROS76].

6. REFERENCES

[BAU78] G.M Baudet,"Asynchronous Iterative Methods," JACM, April 1978.

[CHA69] D. Chazan, and W. Miranker,"Chaotic Relaxation," *Linear Algebra and Applications,* 1969, pp.199-222.

[DUB82] M. Dubois and F.A. Briggs,"Performance of Synchronized Iterative Processes in Multiprocessor Systems," *IEEE Transactions on Software,* Vol. SE-8, No.4, July 1982, pp.419-432.

[KUN76] H.T. Kung,"Synchronized and Asynchronous Parallel Algorithms for Multiprocessors," *Algorithms and Complexity: New Directions and Recent Results,* J.F. Traub Ed., New York: Academic Press, 1976.

[ROB79] J.T. Robinson,"Some Analysis Techniques for Asynchronous Multiprocessor Algorithms," *IEEE Transactions on Software Engineering,* Vol. SE-5, Jan. 1979, pp24-30.

[ROS76] A. Rosenfeld, *et al.* "Scene Labeling by Relaxation Operations," *IEEE Transactions on Systems, Man and Cybernetics,* June 1976.