

## PARTIALLY ASYNCHRONOUS, PARALLEL ALGORITHMS FOR NETWORK FLOW AND OTHER PROBLEMS\*

P. TSENG†, D. P. BERTSEKAS†, AND J. N. TSITSIKLIS†

**Abstract.** The problem of computing a fixed point of a nonexpansive function  $f$  is considered. Sufficient conditions are provided under which a parallel, partially asynchronous implementation of the iteration  $x := f(x)$  converges. These results are then applied to (i) quadratic programming subject to box constraints, (ii) strictly convex cost network flow optimization, (iii) an agreement and a Markov chain problem, (iv) neural network optimization, and (v) finding the least element of a polyhedral set determined by a weakly diagonally dominant, Leontief system. Finally, simulation results illustrating the attainable speedup and the effects of asynchronism are presented.

**Key words.** parallel algorithms, asynchronous algorithms, nonexpansive functions, network flows, neural networks, agreement, Markov chains, Leontief systems

**AMS(MOS) subject classifications.** 49, 90

**1. Introduction.** In this paper we consider the computation of a fixed point of a nonexpansive function  $f$  using parallel, partially asynchronous iterative algorithms of the form  $x := f(x)$ . We give sufficient conditions under which such algorithms converge, we show that some known methods satisfy these conditions, and we propose some new algorithms. The convergence behavior of our methods is qualitatively different from the convergence behavior of most asynchronous algorithms that have been studied in the past by many authors [1]-[3], [5], [8], [27]-[30].

We consider a fixed point problem in the  $n$ -dimensional Euclidean space  $\mathfrak{R}^n$ . We are given functions  $f_i: \mathfrak{R}^n \rightarrow \mathfrak{R}$ ,  $i = 1, \dots, n$ , and we wish to find a point  $x^* \in \mathfrak{R}^n$  such that

$$x^* = f(x^*),$$

where  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  is defined by  $f(x) = (f_1(x), \dots, f_n(x))$ .

We consider a network of processors endowed with local memories, which communicate by message passing, and which do not have access to a global clock. We assume that there are exactly  $n$  processors, each of which maintains its own estimate of a fixed point, and that the  $i$ th processor is responsible for updating  $x_i$ , the  $i$ th component of  $x$ . (If the number of processors is smaller than  $n$ , we may let each processor update several components; the mathematical description of the algorithm does not change and our results apply to this case as well.) We assume that processor  $i$  updates its component by occasionally applying  $f_i$  to its current estimate, say  $x$ , and then transmitting (possibly with some delay) the computed value  $f_i(x)$  to all other processors, which use this value to update the  $i$ th component of their own estimates (see Fig. 1.1).

We use a nonnegative integer variable  $t$  to index the events of interest (e.g., processor updates). We will refer to  $t$  as *time*, although  $t$  need not correspond to the time of a global clock. We use the following notations:

\* Received by the editors November 14, 1988; accepted for publication (in revised form) July 21, 1989.

† Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This work was supported by National Science Foundation grants NSF-ECS-8519058 and NSF-ECS-8552419, with matching funds from Bellcore and Du Pont, and by Army Research Office grant DAAL03-86-K-0171.

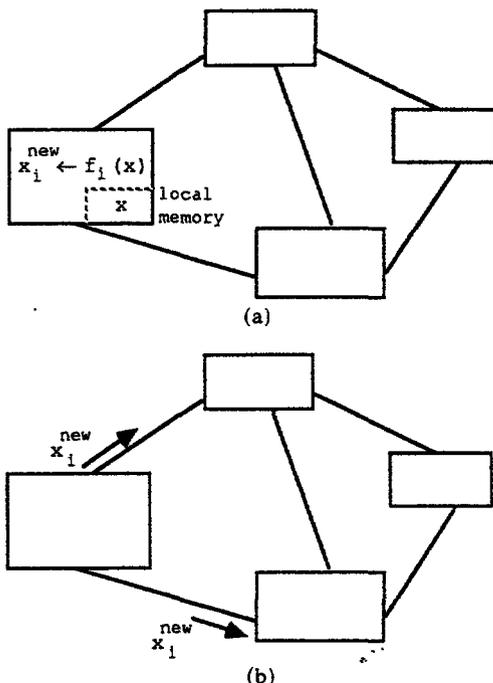


FIG. 1.1. (a) Processor  $i$  computes new estimate of the  $i$ th component of a fixed point. (b) Processor  $i$  transmits new estimate to other processors.

$x_i(t)$  =  $i$ th component of the solution estimate stored by processor  $i$  at time  $t$ .

$\mathcal{T}_i$  = an infinite set of times at which processor  $i$  updates  $x_i$ .

$\tau_{ij}(t)$  = a time at which the  $j$ th component of the solution estimate stored by processor  $i$  at time  $t$  was stored in the local memory of processor  $j$  ( $j = 1, \dots, n$ ;  $t \in \mathcal{T}_i$ ). (Naturally,  $\tau_{ij}(t) \leq t$ .)

In accordance with the above definitions, we postulate that the variables  $x_i(t)$  evolve according to:

$$(1.1) \quad x_i(t+1) = \begin{cases} f_i(x_1(\tau_{i1}(t)), \dots, x_n(\tau_{in}(t))) & \text{if } t \in \mathcal{T}_i, \\ x_i(t) & \text{otherwise.} \end{cases}$$

The initial conditions  $x_i(0)$  are given, and for notational convenience we assume that  $x_i(t) = x_i(0)$  for  $t \leq 0$ , so that the asynchronous iteration (1.1) is well defined for  $\tau_{ij}(t) \leq 0$ . We may view the difference  $t - \tau_{ij}(t)$  as a "communication delay" between the current time  $t$  and the time  $\tau_{ij}(t)$  at which the value of the  $j$ th coordinate, used by processor  $i$  at time  $t$ , was generated at processor  $j$ .

Asynchronous computation models may be divided into *totally asynchronous* and *partially asynchronous*. In the totally asynchronous model [1]–[3], [8], [30], the "delays"  $t - \tau_{ij}(t)$  can become unbounded as  $t$  increases. This is the main difference with the partially asynchronous model, where the amounts  $t - \tau_{ij}(t)$  are assumed bounded; in particular, the following assumption holds.

**Assumption A. (Partial Asynchronism).** There exists a positive integer  $B$  such that, for each  $i$  and each  $t \in \mathcal{T}_i$ , there holds:

- $0 \leq t - \tau_{ij}(t) \leq B - 1$ , for all  $j \in \{1, \dots, n\}$ .
- There exists  $t' \in \mathcal{T}_i$  for which  $1 \leq t' - t \leq B$ .
- $\tau_{ii}(t) = t$ .

Parts (a) and (b) of Assumption A state that both the communication delays and the processor idle periods are bounded and can be expected to hold in most practical cases; for example, (b) holds if each processor uses a local clock, if the ratio of the speeds of different local clocks is bounded, and if each processor computes periodically according to its own local clock (see [7], p. 484). Part (c) of Assumption A states that a processor  $i$  always uses the most recent value of its own component  $x_i$ . This assumption typically holds in practice, but it is interesting to note that, while it is necessary for our results (see the proof of Lemma 2.3(a)), it is not needed in the convergence analysis of totally asynchronous algorithms.

Partially asynchronous iterations have already been studied in the context of gradient optimization algorithms, for which it was shown that convergence is obtained provided that the bound  $B$  of Assumption A is sufficiently small [27]–[29]. Our results concern a fundamentally different class of partially asynchronous methods which are convergent for *every* value of the bound  $B$ . At least two interesting examples of such methods are known: the agreement algorithm of [29] and the Markov chain algorithm of [20]. However, it appears that these methods have not been recognized earlier as a class. Their convergence behavior is somewhat surprising because their totally asynchronous versions do not converge in general; for a counterexample, see [7, p. 484].

In this paper we focus on the convergence issues of partially asynchronous methods with arbitrarily large values of the asynchronism bound  $B$ . Our main result (Proposition 2.1) is the first general convergence result for these methods. In §§ 3–7, we show that Proposition 2.1 applies to a variety of methods for several important problems, including the agreement and Markov chain algorithms mentioned earlier. Some of our convergence results are new, even when they are specialized to the case of synchronous algorithms; for example, the convergence of Jacobi relaxation methods for strictly convex cost network flow problems in § 4.

**2. A general convergence theorem.** Throughout this paper, we let  $X^* = \{x \in \mathfrak{R}^n \mid f(x) = x\}$  be the set of fixed points of  $f$  and, for each  $x \in \mathfrak{R}^n$ , we let  $\|x\| = \max_{i=1, \dots, n} |x_i|$  denote the maximum norm of  $x$ . For any  $x \in \mathfrak{R}^n$ , we denote by  $\rho(x)$  the distance of  $x$  from  $X^*$ , defined by

$$\rho(x) = \inf_{y \in X^*} \|x - y\|.$$

Finally, given any  $x \in \mathfrak{R}^n$  and  $x^* \in X^*$ , we let  $I(x; x^*)$  be the set of indices of coordinates of  $x$  that are farthest away from  $x^*$ , that is,

$$I(x; x^*) = \{i \mid |x_i - x_i^*| = \|x - x^*\|\},$$

and we also denote

$$U(x; x^*) = \{y \in \mathfrak{R}^n \mid y_i = x_i \text{ for all } i \in I(x; x^*),$$

$$\text{and } |y_i - x_i^*| < \|x - x^*\| \text{ for all } i \notin I(x; x^*)\}.$$

Loosely speaking,  $U(x; x^*)$  is the set of all vectors  $y$  with  $\|y - x^*\| = \|x - x^*\|$  that agree with  $x$  in the components that are farthest away from  $x^*$  (see Fig. 2.1).

Our main assumption on the structure of  $f$  is the following.

**Assumption B.**

(a)  $f$  is continuous.

(b) The set of fixed points  $X^*$  is convex and nonempty.

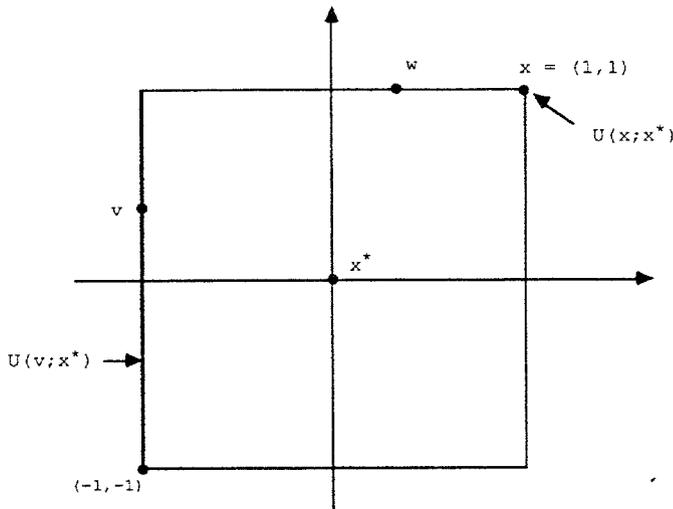


FIG. 2.1. Illustration of the sets  $I(\cdot; x^*)$  and  $U(\cdot; x^*)$ . Let  $n=2$  and suppose that  $x^*=(0,0) \in X^*$ . For the indicated points  $x$ ,  $v$ , and  $w$ , we have  $I(x; x^*)=\{1, 2\}$ ,  $I(v; x^*)=\{1\}$ ,  $I(w; x^*)=\{2\}$ . The set  $U(v; x^*)$  is the set of all vectors of the form  $(-1, c)$ , where  $c$  satisfies  $-1 < c < 1$ , which is the segment joining the points  $(-1, -1)$  and  $(-1, 1)$ , the endpoints excluded. Similarly,  $U(w; x^*)=\{(c, 1) \mid -1 < c < 1\}$ . Finally, we have  $U(x, x^*)=\{x\}$ .

(c)  $\|f(x) - x^*\| \leq \|x - x^*\|$ , for all  $x \in \mathfrak{R}^n$ , for all  $x^* \in X^*$ .

(d) For every  $x \in \mathfrak{R}^n$  and  $x^* \in X^*$  such that  $\|x - x^*\| = \rho(x) > 0$ , there exists some  $i \in I(x; x^*)$  such that  $f_i(y) \neq y_i$  for all  $y \in U(x; x^*)$ .

Part (c) of Assumption B states that  $f$  does not increase the distance from a fixed point and will be referred to as the *pseudo-nonexpansive* property. This is slightly weaker than requiring that  $f$  be nonexpansive (that is,  $\|f(x) - f(y)\| \leq \|x - y\|$  for all  $x$  and  $y$  in  $\mathfrak{R}^n$ ) and in certain cases is easier to verify (see § 4). We interpret part (d) as follows: Consider some  $x \notin X^*$ . Then  $f(x) \neq x$ , and there exists some  $i$  such that  $f_i(x) \neq x_i$ . Assumption B(d) imposes the additional requirement that such an  $i$  can be found among the set of worst indices, that is,  $i$  belongs to the set  $I(x; x^*)$  of indices corresponding to components farthest away from a closest element of  $X^*$ . Furthermore, if we change some of the other components of  $x$  to obtain another vector  $y \in U(x; x^*)$ , we still retain the property  $f_i(y) \neq y_i$ , for this particular  $i$ . This part of Assumption B is usually the most difficult to verify in specific applications.

Unfortunately, the following simple example shows that Assumptions A and B alone are not sufficient for convergence of even the synchronous version of iteration (1.1): Suppose that  $f(x_1, x_2) = (x_2, x_1)$  (which can be verified to satisfy Assumption B with  $X^* = \{(\lambda, \lambda) \mid \lambda \in \mathfrak{R}\}$ ). Then the sequence  $\{x(t)\}$  generated by the synchronous iteration  $x(t+1) = f(x(t))$  (which is a special case of (1.1)), with  $x(0) = (1, 0)$ , oscillates between  $(1, 0)$  and  $(0, 1)$ .

The difficulty in this example is that, at each iteration, while the worst coordinate  $i \in I(x; x^*)$  is changed from 1 to 0, the other coordinate is increased from 0 to 1, and the distance  $\rho(x)$  from  $X^*$  is not changed. The following assumption is designed to prevent such behavior.

**Assumption C.** For any  $i$ ,  $x \in \mathfrak{R}^n$ , and  $x^* \in X^*$ , if  $f_i(x) \neq x_i$ , then  $|f_i(x) - x_i| < \|x - x^*\|$ .

An important fact, shown below, is that any mapping satisfying Assumption B can be modified by introducing a relaxation parameter, so that it satisfies Assumption C as well.

LEMMA 2.1. *Let  $h : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  be a function satisfying Assumption B. Then the mapping  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  whose  $i$ th component is*

$$f_i(x) = (1 - \gamma_i)x_i + \gamma_i h_i(x),$$

where  $\gamma_1, \dots, \gamma_n$  are scalars in  $(0, 1)$ , has the same set of fixed points as  $h$  and satisfies both Assumptions B and C.

*Proof.* It is easily seen that  $f$  is continuous and has the same set of fixed points as  $h$ , so it satisfies parts (a) and (b) of Assumption B. Since  $f_i(x) \neq x_i$  if and only if  $h_i(x) \neq x_i$ , we see that  $f$  satisfies part (d) of Assumption B. Since  $h$  is pseudo-nonexpansive, for all  $i, x \in \mathfrak{R}^n$ , and  $x^* \in X^*$ , both  $x_i$  and  $h_i(x)$  belong to the interval

$$[x_i^* - \|x - x^*\|, x_i^* + \|x - x^*\|].$$

Therefore,  $f_i(x)$ , which is a convex combination of  $x_i$  and  $h_i(x)$ , must also belong to this interval, proving that  $f$  is pseudo-nonexpansive, (cf. part (c) of Assumption B). Furthermore, if  $h_i(x) \neq x_i$ , then the convex combination  $f_i(x)$  must belong to the interior of this interval, showing that  $f$  satisfies Assumption C.  $\square$

We now prove our main convergence result, showing that Assumptions A, B, and C are sufficient for the sequence  $\{x(t)\}$  generated by the asynchronous iteration (1.1) to converge to an element of  $X^*$ . To motivate our proof, consider the synchronous iteration  $x(t+1) = f(x(t))$ . Under Assumptions B and C, either (i)  $\rho(x(t+1)) < \rho(x(t))$  or (ii)  $\rho(x(t+1)) = \rho(x(t))$  and  $x(t+1)$  has a smaller number of components at a distance  $\rho(x(t))$  from  $X^*$  than  $x(t)$ . Thus, case (ii) can occur for at most  $n$  successive iterations before case (i) occurs. This argument can be extended for the asynchronous iteration (1.1), but because of communication and computation delays (each bounded by  $B$ , due to Assumption A), the number of time steps until the distance to  $X^*$  decreases is upper bounded by roughly  $2nB$  (see part (c) of Lemma 2.3).

PROPOSITION 2.1. *Suppose that  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  satisfies Assumptions B and C, and suppose that Assumption A (partial asynchronism) holds. Then the sequence  $\{x(t)\}$  generated by the asynchronous iteration (1.1) converges to some element of  $X^*$ .*

*Proof.* For each integer  $t \geq 0$  denote

$$z(t) = (x(t - B + 1), \dots, x(t)),$$

$$d(z(t)) = \min_{x^* \in X^*} \{ \max \{ \|x(t - B + 1) - x^*\|, \dots, \|x(t) - x^*\| \} \}.$$

Notice that the minimum in the definition of  $d(z(t))$  is attained because the set  $X^*$  is closed (as a consequence of the continuity of  $f$ ). For each  $t \geq 0$ , we fix an element  $x^*(t)$  of  $X^*$  attaining the minimum

$$(2.1) \quad x^*(t) = \arg \min_{x^* \in X^*} \{ \max \{ \|x(t - B + 1) - x^*\|, \dots, \|x(t) - x^*\| \} \}.$$

As part of the proof of Proposition 2.1, we prove some preliminary facts in the following two lemmas, which show that the distance  $d(z(t))$  cannot increase at any iteration while it decreases strictly "every few" iterations.

LEMMA 2.2.  *$d(z(t+1)) \leq d(z(t))$ , for all  $z(t) \in \mathfrak{R}^{nB}$ , for all  $t \geq 0$ .*

*Proof.* We will prove by induction that

$$(2.2) \quad \|x(r) - x^*(t)\| \leq d(z(t)), \quad \forall r \geq t - B + 1,$$

which implies the result. From (2.1) and the definition of  $d(z(t))$ , this inequality holds for  $r \in \{t - B + 1, \dots, t\}$ . Suppose that it holds for all  $r \in \{t - B + 1, \dots, r'\}$ , where  $r'$  is some integer greater than or equal to  $t$ . We will show that it holds for  $r' + 1$ . By (1.1), for each  $i$ , either  $x_i(r' + 1) = x_i(r')$  or  $x_i(r') = f_i(x_i(\tau_{i1}(r')), \dots, x_n(\tau_{in}(r')))$ . In the former case, we have  $|x_i(r' + 1) - x_i^*(t)| = |x_i(r') - x_i^*(t)| \leq d(z(t))$  by the induction hypothesis. In the latter case, we have by Assumption A(a),  $r' - B + 1 \leq \tau_{ij}(r') \leq r'$ , so by the induction hypothesis,  $|x_j(\tau_{ij}(r')) - x_j^*(t)| \leq d(z(t))$  for all  $j$ . Using the pseudo-nonexpansive property of Assumption B(c), we obtain

$$|x_i(r' + 1) - x_i^*(t)| \leq \max_j |x_j(\tau_{ij}(r')) - x_j^*(t)| \leq d(z(t)).$$

Thus, in either case we have  $|x_i(r' + 1) - x_i^*(t)| \leq d(z(t))$ , and this is true for every index  $i$ . Therefore,  $\|x(r' + 1) - x^*(t)\| \leq d(z(t))$ , completing the induction.  $\square$

LEMMA 2.3. Fix some  $t \geq 0$  for which  $d(z(t)) > 0$  and denote

$$J(r) = \{i \mid |x_i(r) - x_i^*(t)| = d(z(t))\}, \quad \forall r \geq t.$$

(a) If  $x_i(r + 1) \neq x_i(r)$  for some  $r \geq t$ , then  $i \notin J(r + 1)$ .

(2.3) (b)  $J(r + 1) \subseteq J(r)$ , for all  $r \geq t$ .

(c)  $d(z(t + 2nB + B - 1)) < d(z(t))$ .

*Proof.* For convenience, we will use the notation

$$\beta = d(z(t)), \quad x^* = x^*(t).$$

(a) If  $x_i(r + 1) \neq x_i(r)$ , we have  $r \in \mathcal{T}_i$ . Furthermore,

$$f_i(x_i(\tau_{i1}(r)), \dots, x_n(\tau_{in}(r))) = x_i(r + 1) \neq x_i(r) = x_i(\tau_{ii}(r)),$$

where the last equality follows from Assumption A(c). Using Assumption C, we obtain

$$|x_i(r + 1) - x_i^*| < \max_j |x_j(\tau_{ij}(r)) - x_j^*| \leq \beta,$$

where the last inequality follows from  $r - B + 1 \leq \tau_{ij}(r) \leq r$  (cf. Assumption A(a)) and Lemma 2.2 (cf., (2.2)). Thus,  $i \notin J(r + 1)$ .

(b) If  $i \in J(r + 1)$ , then part (a) shows that  $x_i(r) = x_i(r + 1)$ , which implies that  $i \in J(r)$ .

(c) We first show by contradiction that, for all  $r \geq t$ ,

$$(2.4) \quad d(z(r + 2B)) = \beta \Rightarrow J(r + 2B) \neq J(r).$$

Suppose that, for some  $r \geq t$ , we have  $d(z(r + 2B)) = \beta$  and  $J(r) = J(r + 2B)$ . By part (b),  $J(r) = J(r + 1) = \dots = J(r + 2B)$ . Denote  $J = J(r)$ . Then, by part (a),

$$(2.5) \quad x_i(r) = x_i(r + 1) = \dots = x_i(r + 2B), \quad \forall i \in J,$$

and by the definition of  $J$ ,

$$(2.6) \quad |x_i(r) - x_i^*| < \beta, \dots, |x_i(r + 2B) - x_i^*| < \beta, \quad \forall i \in J.$$

Now, from the definition of  $J$ ,  $x^*$  and  $\beta$  we have that  $|x_i(r) - x_i^*| = \beta$  for all  $i \in J$ ; hence (2.6) implies

$$(2.7) \quad \|x(r) - x^*\| = \beta, \quad J = I(x(r); x^*).$$

Also by Assumption A(b), for each  $i \in J$ , there exists  $r_i \in \{r+B, \dots, r+2B-1\}$  such that  $r_i \in \mathcal{F}_i$  and the iteration (1.1) yields

$$(2.8) \quad x_i(r_i+1) = f_i(x_1(\tau_{i1}(r_i)), \dots, x_n(\tau_{in}(r_i))), \quad \forall i \in J.$$

Let us denote

$$x' = (x_1(\tau_{i1}(r_i)), \dots, x_n(\tau_{in}(r_i))), \quad \forall i \in J.$$

By Assumption A(c),  $\tau_{ii}(r_i) = r_i$  for all  $i \in J$ , which together with (2.5) implies that

$$x_i(r_i+1) = x_i(\tau_{ii}(r_i)), \quad \forall i \in J.$$

Therefore, (2.8) can be written as

$$(2.9) \quad x_i^i = f_i(x^i), \quad \forall i \in J.$$

Furthermore, by Assumption A(a),  $r \leq \tau_{ij}(r_i) \leq r+2B$  for all  $i \in J$  and all  $j$ , which together with (2.5)–(2.6) implies that

$$\begin{aligned} x_j^i &= x_j(r), \quad \forall i \in J, \quad \forall j \in J, \\ |x_j^i - x_j^*| &< \beta, \quad \forall i \in J, \quad \forall j \notin J. \end{aligned}$$

Therefore from (2.7) we also have

$$(2.10) \quad x' \in U(x(r); x^*), \quad \forall i \in J.$$

It now follows that

$$\beta = \|x(r) - x^*\| > \rho(x(r)),$$

since if  $\|x(r) - x^*\| = \rho(x(r))$ , then in view of the fact  $I(x(r); x^*) = J$  (cf. (2.7)) and (2.9)–(2.10), Assumption B(d) would be violated.

Thus, we conclude that there exist  $y^* \in X^*$  and  $\theta \in [0, \beta)$  such that  $\|x(r) - y^*\| = \theta$ . Let

$$\begin{aligned} \varepsilon &= \max \{|x_i(m) - x_i^*| \mid i \notin J, m = r+B, \dots, r+2B-1\}, \\ M &= \max \{|x_i(m) - y_i^*| \mid i \notin J, m = r+B, \dots, r+2B-1\} \end{aligned}$$

(see Fig. 2.2). Since  $X^*$  is convex, we have that, for any  $\omega \in (0, 1)$ ,  $z^* = (1-\omega)x^* + \omega y^*$

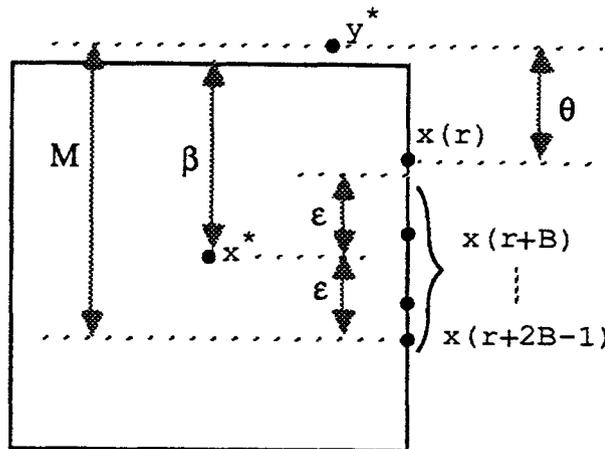


FIG. 2.2

is in  $X^*$  and, for  $m = r + B, \dots, r + 2B - 1$ ,

$$\begin{aligned} |x_i(m) - z_i^*| &= |x_i(r) - z_i^*| \\ &\leq (1 - \omega)|x_i(r) - x_i^*| + \omega|x_i(r) - y_i^*| \\ &= (1 - \omega)\beta + \omega\theta, \quad \forall i \in J, \\ |x_i(m) - z_i^*| &\leq (1 - \omega)|x_i(m) - x_i^*| + \omega|x_i(m) - y_i^*| \\ &\leq (1 - \omega)\varepsilon + \omega M, \quad \forall i \notin J. \end{aligned}$$

Since  $\varepsilon < \beta$  and  $\theta < \beta$ , we have that, for  $\omega$  sufficiently small,

$$\|x(m) - z^*\| < \beta, \quad \forall m = r + B, \dots, r + 2B - 1.$$

This implies that  $d(z(r + 2B - 1)) < \beta$ , a contradiction.

Since by Lemma 2.2,  $d(z(r))$  is nonincreasing, either  $d(z(t + 2nB - 1)) < \beta$ , in which case the result is proved, or  $d(z(t + 2nB - 1)) = \beta$ . In the latter case, by (2.3) and (2.4),  $J(t + 2nB) = \dots = J(t + 2nB + B - 1) = \emptyset$ , and

$$d(z(t + 2nB + B - 1)) = \max \{ \|x(t + 2nB) - x^*\|, \dots, \|x(t + 2nB + B - 1) - x^*\| \} < \beta. \quad \square$$

We now complete the proof of Proposition 2.1.

By (2.2), the sequence  $\{z(t)\}$  is bounded and, by Lemma (2.3)(c),  $d(z(t))$  monotonically decreases to some limit  $\beta$ . If  $\beta = 0$ , then Lemma 2.2 and (2.2) imply that  $\{x(r)\}$  has a unique limit point, which is in  $X^*$ , and our proof is complete. Suppose, to obtain a contradiction, that  $\beta > 0$ . Let

$$\Delta t = 2nB + B - 1.$$

Since, by (2.2),  $\{z(t)\}$  is bounded, there exist some  $z^* \in \mathfrak{R}^{nB}$ ,  $z^{**} \in \mathfrak{R}^{nB}$  and a subsequence  $T$  of  $\{0, 1, \dots\}$  such that

$$(2.11) \quad \{z(t)\}_{t \in T} \rightarrow z^*, \quad \{z(t + \Delta t)\}_{t \in T} \rightarrow z^{**}.$$

Note that since  $d(z(t)) \rightarrow \beta$  and  $d$  is a continuous function, (2.11) implies that  $d(z^*) = d(z^{**}) = \beta$ .

From (1.1), Assumption A and the definition of  $z(t)$ , we see that we can express  $z(t + \Delta t)$  as a continuous function of  $z(t)$ . In particular, we can write

$$(2.12) \quad z(t + \Delta t) = g(z(t); \Gamma(t)),$$

where  $\Gamma(t) = (\Gamma_1(t), \dots, \Gamma_n(t))$  and  $\Gamma_i(t)$  denotes the set

$$(2.13) \quad \Gamma_i(t) = \{(r - t, \tau_{i1}(r) - t, \dots, \tau_{in}(r) - t) \mid r \in \mathcal{S}_i \cap \{t, \dots, t + \Delta t\}\},$$

and  $g(\cdot; \Gamma(t)): \mathfrak{R}^{nB} \rightarrow \mathfrak{R}^{nB}$  is some continuous function that depends on  $f$  and  $\Gamma(t)$  only. (Note that  $g(\cdot; \Gamma(t))$  is the composition of the  $f_i$ 's in an order determined by  $\Gamma(t)$  and is continuous because  $f$  is continuous.) Since (cf. (2.13) and Assumption A)  $\Gamma(t)$  takes values from a finite set, by further passing into a subsequence, if necessary, we can assume that  $\Gamma(t)$  is the same set for all  $t \in T$ . Let  $\Gamma = (\Gamma_1, \dots, \Gamma_n)$  denote this set. Then from (2.12) we obtain that

$$z(t + \Delta t) = g(z(t); \Gamma), \quad \forall t \in T.$$

Since  $g(\cdot; \Gamma)$  is continuous, this, together with (2.11), implies that  $z^{**} = g(z^*; \Gamma)$  or, equivalently,  $z(\Delta t) = z^{**}$  if  $z(0) = z^*$  and

$$\{(r, \tau_{i1}(r), \dots, \tau_{in}(r)) \mid r \in \mathcal{S}_i \cap \{0, \dots, \Delta t\}\} = \Gamma_i, \quad \forall i.$$

Since  $d(z^*) = \beta > 0$ , this, together with Lemma 2.3(c), implies that  $d(z^{**}) < d(z^*)$ , contradicting the hypothesis  $d(z^{**}) = \beta$ .  $\square$

The convexity of  $X^*$  is sometimes hard to verify. For this reason we will consider another assumption that is stronger than Assumption B but is easier to verify.

**Assumption B'.**

- (a)  $f$  is continuous.
- (b) The set of fixed points  $X^*$  is nonempty.
- (c)  $\|f(x) - x^*\| \leq \|x - x^*\|$ , for all  $x \in \mathfrak{R}^n$ , for all  $x^* \in X^*$ .
- (d) For every  $x \notin X^*$  and  $x^* \in X^*$ , there exists some  $i \in I(x; x^*)$  such that  $f_i(y) \neq y_i$  for all  $y \in U(x; x^*)$  such that  $y \notin X^*$ .

Compared to Assumption B, part (d) of the new assumption is stronger but part (b) is weaker because convexity is not assumed. We have the following result.

**LEMMA 2.4.** *Assumption B' implies Assumption B.*

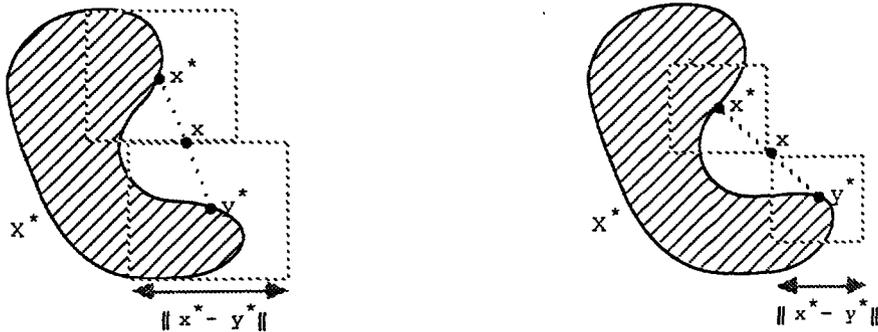
*Proof.* It can be seen that Assumption B'(d) implies Assumption B(d), so we only need to show that  $X^*$  is convex. Suppose the contrary. Since  $X^*$  is closed, then there exist  $x^* \in X^*$  and  $y^* \in X^*$  such that  $(x^* + y^*)/2 \notin X^*$ . Let  $x = (x^* + y^*)/2$ . It can be seen that  $\|x - x^*\| = \|x - y^*\| > 0$ ,  $x \notin X^*$ , and  $I(x; x^*) = I(x; y^*)$  (see Fig. 2.3). By Assumption B'(d), there exists  $i \in I(x; x^*)$  such that  $f_i(x) \neq x_i$ . Suppose that  $x_i > y_i^*$ . Then if  $f_i(x) > x_i$ , we obtain  $\|f(x) - y^*\| \geq f_i(x) - y_i^* > x_i - y_i^* = \|x - y^*\|$  and if  $f_i(x) < x_i$ , we similarly obtain  $\|f(x) - x^*\| > \|x - x^*\|$ . In either case Assumption B'(c) is contradicted. The case where  $x_i < y_i^*$  is treated analogously.  $\square$

Assumption B will be used in § 4, while Assumption B' will be used in §§ 3, 6, and 7.

**3. Nonexpansive mappings on a box.** Let  $g: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  be a continuously differentiable function satisfying the following assumption:

**Assumption D.**

- (a) For each  $i$ ,  $\sum_{j=1}^n |\partial g_j(x) / \partial x_j| \leq 1$ , for all  $x \in \mathfrak{R}^n$ .
- (b) For each  $i$  and  $j$ , either  $\partial g_i(x) / \partial x_j = 0$ , for all  $x \in \mathfrak{R}^n$ , or  $\partial g_i(x) / \partial x_j \neq 0$ , for all  $x \in \mathfrak{R}^n$ .



$I(x; x^*) = I(x; y^*) = \{2\}.$

$I(x; x^*) = I(x; y^*) = \{1, 2\}.$

FIG. 2.3. Two configurations of  $x^*$  and  $y^*$ .

(c) The graph with node set  $\{1, \dots, n\}$  and arc set  $\{(i, j) | \partial g_i(x) / \partial x_j \neq 0\}$  is strongly connected.

Let  $C$  be a box (possibly unbounded) in  $\mathfrak{R}^n$ , i.e.,

$$C = \{x \in \mathfrak{R}^n | l_i \leq x_i \leq c_i, \forall i\},$$

for some scalars  $l_i$  and  $c_i$  satisfying  $l_i \leq c_i$  (we allow  $l_i = -\infty$  or  $c_i = +\infty$ ). Let also  $[x]^+$



Since  $h$  is also continuous and  $Y$  is convex and compact, a theorem of Brouwer ([11], p. 17) shows that  $h$  has a fixed point. Now suppose  $C$  is bounded. Since  $h(x) \in C$  for all  $x \in C$  and  $C$  is convex and compact, the same theorem of Brouwer shows that  $h$  has a fixed point. Thus, part (b) of Assumption B' is satisfied.

We finally show that Assumption B'(d) holds. Suppose the contrary. Then there exists some  $x \notin X^*$  and some  $x^* \in X^*$ , such that for every  $i \in I(x; x^*)$  there is an  $x^i \in U(x; x^*)$  with  $x^i \notin X^*$  and  $h_i(x^i) = x^i$ . Let  $J = I(x; x^*)$ ,  $\beta = \|x - x^*\|$  and fix some  $i \in J$ . By the Mean Value Theorem, there exists some  $\xi \in \mathfrak{R}^n$  such that  $g_i(x^i) - g_i(x^*) = (\nabla g_i(\xi))^T(x^i - x^*)$ . Let  $a_j = \partial g_i(\xi)/\partial x_j$ . Then

$$\begin{aligned} \beta &= |x_i^i - x_i^*| = |h_i(x^i) - h_i(x^*)| \\ &\leq |g_i(x^i) - g_i(x^*)| \\ &= \left| \sum_j a_j(x_j^i - x_j^*) \right| \\ &\leq \left( \sum_{j \in J} |a_j| \right) \beta + \left( \sum_{j \notin J} |a_j| |x_j^i - x_j^*| \right) \\ &\leq \beta + \sum_{j \notin J} |a_j| (|x_j^i - x_j^*| - \beta), \end{aligned}$$

where the first inequality follows from the fact that the projection onto  $[l_i, c_i]$  is nonexpansive and the last inequality follows from the fact (cf. Assumption D(a)) that  $\sum_j |a_j| \leq 1$ . Since  $|x_j^i - x_j^*| < \beta$  for all  $j \notin J$ , the above inequality implies that  $a_j = 0$  for all  $j \notin J$ . Since the choice of  $i \in J$  was arbitrary, we obtain from Assumption D(b) that  $\partial g_i(\xi)/\partial x_j = 0$  for all  $\xi \in \mathfrak{R}^n$ ,  $i \in J$ ,  $j \notin J$ . By Assumption D(c), we must have that  $J = \{1, \dots, n\}$ . In that case,  $U(x; x^*)$  is a singleton and all the vectors  $x^i$  are equal. It then follows from the equalities  $h_i(x^i) = x^i$ , for all  $i$ , that each  $x^i$  is a fixed point of  $h$ , a contradiction of the hypothesis  $x^i \notin X^*$ .  $\square$

Since Assumption B' is satisfied, the partially asynchronous iteration

$$x := (1 - \gamma)x + \gamma[g(x)]^+$$

(with  $0 < \gamma < 1$ ) converges (cf. Lemmas 2.1, 2.4, and Proposition 2.1).

An important special case is obtained if  $C = \mathfrak{R}^n$ ,  $g(x) = Ax + b$ , where  $A$  is an  $n \times n$  matrix and  $b$  is a given vector in  $\mathfrak{R}^n$ . Thus, the problem is to solve the linear system

$$x = Ax + b,$$

and Assumption D amounts to the requirement that  $A = [a_{ij}]$  is irreducible (see [22] for a definition of irreducibility) and  $\sum_j |a_{ij}| \leq 1$ , for all  $i$ . Then, provided that the system  $x = Ax + b$  has a solution (not necessarily unique), the partially asynchronous iteration

$$x := (1 - \gamma)x + \gamma(Ax + b)$$

(with  $0 < \gamma < 1$ ) will converge to such a solution.

As a special case of our results, we obtain convergence of the synchronous iteration

$$x(t+1) = (1 - \gamma)x(t) + \gamma(Ax(t) + b).$$

This seems to be a new result under our assumptions. Previous convergence results [17], [22] have made the stronger assumption that either: (a)  $A$  is irreducible and  $\sum_j |a_{ij}| \leq 1$ , for all  $i$ , with strict inequality for at least one  $i$ , or (b)  $\sum_j |a_{ij}| < 1$ , for all  $i$ . Two other important special cases are studied below.

**3.1. Quadratic costs subject to box constraints.** Consider the following problem.

$$(3.3) \quad \begin{array}{ll} \text{Minimize} & x^T Q x / 2 + p^T x \\ \text{Subject to} & x \in C, \end{array}$$

where  $Q = [q_{ij}]$  is a symmetric, irreducible, nonnegative definite matrix of dimension  $n \times n$  satisfying the weak diagonal dominance condition

$$(3.4) \quad \sum_{j \neq i} |q_{ij}| \leq q_{ii}, \quad q_{ii} > 0, \quad \forall i,$$

$p$  is an element of  $\mathfrak{R}^n$ , and  $C$  is, as before, a box in  $\mathfrak{R}^n$ .

Let  $D$  denote the diagonal matrix whose  $i$ th diagonal entry is  $q_{ii}$ . Let  $A = I - D^{-1}Q$  and  $b = -D^{-1}p$ . We have the following result.

**PROPOSITION 3.2.** *The function  $g: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  defined by  $g(x) = Ax + b$  satisfies Assumption D.*

*Proof.*  $g$  is clearly continuously differentiable and (cf. (3.4))  $\sum_j |a_{ij}| = \sum_{j \neq i} |q_{ij}| / q_{ii} \leq 1$  for all  $i$ . Since  $\partial g_i(x) / \partial x_j = a_{ij}$  for all  $x \in \mathfrak{R}^n$  and  $A$  is irreducible,  $g$  satisfies Assumption D.  $\square$

It can be seen (by using the Kuhn-Tucker optimality conditions [23]) that each optimal solution of (3.3) is a fixed point of  $[Ax + b]^+$  and vice versa, where  $[\cdot]^+$  denotes the orthogonal projection onto  $C$ . Hence, if (3.3) has an optimal solution, then (cf. Lemma 2.1, 2.4, and Propositions 2.1, 3.1, 3.2) the partially asynchronous iteration

$$(3.5) \quad x := (1 - \gamma)x + \gamma[Ax + b]^+$$

(with  $0 < \gamma < 1$ ) converges to such a solution. Note that for  $\gamma = 1$ , the iteration (3.5) takes the form  $x := [x - D^{-1}(Qx + p)]^+$  which is a diagonally scaled gradient projection iteration. However, this iteration need not be convergent in the absence of additional assumptions.

**3.2. Separable quadratic costs with sparse 0, +1, -1 matrix.** Consider the following problem.

$$(3.6) \quad \begin{array}{ll} \text{Minimize} & w^T D w / 2 + \beta^T w \\ \text{Subject to} & E w \geq d, \end{array}$$

where  $D$  is an  $m \times m$  positive definite diagonal matrix,  $\beta$  is an element of  $\mathfrak{R}^m$ ,  $d$  is an element of  $\mathfrak{R}^n$ , and  $E = [e_{ik}]$  is an  $n \times m$  matrix having at most two nonzero entries per column, and each nonzero entry is either -1 or 1. Furthermore, we assume that the undirected graph  $\mathcal{G}$  with node set  $\{1, \dots, n\}$  and arc set  $\{(i, j) | e_{ik} \neq 0 \text{ and } e_{jk} \neq 0 \text{ for some } k\}$  is connected.

Consider the following Lagrangian dual [23] of (3.6).

$$\begin{array}{ll} \text{Minimize} & x^T Q x / 2 + p^T x \\ \text{Subject to} & x \geq 0, \end{array}$$

where  $Q = ED^{-1}E^T$ ,  $p = -d - ED^{-1}\beta$ . We show below that this is a special case of the problem considered in the previous subsection.

**PROPOSITION 3.3.**  *$Q$  is symmetric, irreducible, nonnegative definite and weakly diagonally dominant (cf. (3.4)).*

*Proof.* Since  $D$  is symmetric and positive definite,  $Q$  is symmetric and nonnegative definite. To see that  $Q$  satisfies (3.4), let  $\alpha_k$  denote the  $k$ th diagonal entry of  $D$  ( $\alpha_k > 0$ ), let  $O(i)$  denote the set of indices  $k$  such that  $e_{ik} \neq 0$ , and let  $q_{ij}$  denote the  $(i, j)$ th entry

of  $Q$ . Then

$$|q_{ij}| = \left| \sum_k e_{ik} (\alpha_k)^{-1} e_{jk} \right| \leq \sum_{k \in O(i) \cap O(j)} (\alpha_k)^{-1}$$

with equality holding if  $i = j$ . Hence, for each  $i$ ,

$$\begin{aligned} \sum_{j \neq i} |q_{ij}| &\leq \sum_{j \neq i} \sum_{k \in O(i) \cap O(j)} (\alpha_k)^{-1} \\ &\leq \sum_{k \in O(i)} (\alpha_k)^{-1} \\ &= q_{ii}, \end{aligned}$$

where the second inequality follows from the fact that if  $k \in O(i) \cap O(j)$  for some  $j$ , then  $k \notin O(i) \cap O(j')$  for all  $j'$  not equal to  $i$  or  $j$ . Finally,  $Q$  is irreducible because  $\mathcal{G}$  is connected and  $q_{ij} \neq 0$  for  $i \neq j$  if and only if there exists some  $k$  such that  $e_{ik} \neq 0$  and  $e_{jk} \neq 0$ .  $\square$

An example of constraints  $Ew \geq d$  satisfying our conditions on  $E$  is

$$\sum_k w_k \leq 1 \quad \text{and} \quad \sum_{k \in K_r} w_k \geq 0 \quad \text{for } r = 1, 2, \dots, R,$$

where  $K_1, K_2, \dots, K_R$  are some mutually disjoint subsets of  $\{1, 2, \dots, m\}$ . Such constraints often arise in resource allocation problems.

**4. Strictly convex cost network flow problems.** Consider a connected, directed graph (network) with the set of nodes  $\mathcal{N} = \{1, \dots, n\}$  and the set of arcs  $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ . We assume that  $i \neq j$  for every arc  $(i, j)$  and that at most one arc connects any ordered pair of nodes, so that the arc  $(i, j)$  has unambiguous meaning. (These restrictions can be easily removed.) For each node  $i \in \mathcal{N}$ , denote by  $\mathcal{D}(i)$  the set of downstream neighbors of  $i$  (that is,  $\mathcal{D}(i) = \{j \mid (i, j) \in \mathcal{A}\}$ ) and by  $\mathcal{U}(i)$  the set of upstream neighbors of  $i$  (that is,  $\mathcal{U}(i) = \{j \mid (j, i) \in \mathcal{A}\}$ ). Consider the following problem:

(4.1) Minimize  $\sum_{(i,j) \in \mathcal{A}} a_{ij}(f_{ij})$

(4.2) Subject to  $\sum_{j \in \mathcal{D}(i)} f_{ij} - \sum_{j \in \mathcal{U}(i)} f_{ji} = s_i, \quad \forall i \in \mathcal{N}$ ,

where each  $a_{ij} : \mathfrak{R} \rightarrow (-\infty, +\infty]$  is a strictly convex, lower semicontinuous function and each  $s_i$  is a real number. We interpret  $f_{ij}$  as the flow on the arc  $(i, j)$ ,  $s_i$  as the supply (or demand if  $s_i < 0$ ) at node  $i$ , and  $a_{ij}(f_{ij})$  as the cost of sending a flow of  $f_{ij}$  on arc  $(i, j)$ . The goal is then to find a set of arc flows that minimizes the total cost while satisfying the flow conservation constraints (4.2) (see Fig. 4.1). Note that capacity constraints of the form

$$b_{ij} \leq f_{ij} \leq c_{ij},$$

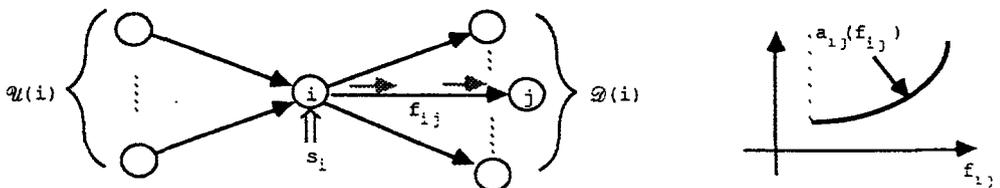


FIG. 4.1

where  $b_{ij}$ ,  $c_{ij}$  are given scalars, can be incorporated into the cost function  $a_{ij}$  by letting  $a_{ij}(f_{ij}) = +\infty$  for  $f_{ij} \notin [b_{ij}, c_{ij}]$ .

The above network flow problem is an important optimization problem, with applications to data networks, traffic assignment, matrix balancing, etc. The interested reader is referred to [7, Chap. 5] for a detailed discussion of this problem. (Also see [5], [6], [9], [12], [21], [24], [31]–[33].)

Denote by  $g_{ij}: \mathfrak{R} \rightarrow (-\infty, +\infty]$  the *conjugate function* ([23, § 12]; [24, p. 330]) of  $a_{ij}$ , i.e.,

$$(4.3) \quad g_{ij}(\eta) = \sup_{\zeta \in \mathfrak{R}} \{\zeta\eta - a_{ij}(\zeta)\}.$$

Each  $g_{ij}$  is convex and, by assigning a Lagrange multiplier  $p_i$  (also called a *price*) to the  $i$ th constraint of (4.2), we can formulate the dual problem ([24, § 8G]) of (4.1) as the following convex minimization problem.

$$(4.4) \quad \begin{aligned} \text{Minimize} \quad & q(p) = \sum_{(i,j) \in \mathcal{A}} g_{ij}(p_i - p_j) - \sum_{i \in \mathcal{N}} p_i s_i \\ \text{Subject to} \quad & p \in \mathfrak{R}^n. \end{aligned}$$

We make the following assumption.

*Assumption E.*

(a) Each conjugate function  $g_{ij}$  is real valued.

(b) The set  $P^*$  of optimal solutions of the dual problem (4.4) is nonempty.

Assumption E implies (cf. [24, § 11D]) that the original problem (4.1) has an optimal solution, and the optimal objective value for (4.1) and (4.4) sum to zero. Furthermore, the strict convexity of the  $a_{ij}$ 's implies that (4.1) has a *unique* optimal solution, which we denote by  $f^* = (\dots, f_{ij}^*, \dots)_{(i,j) \in \mathcal{A}}$ , and that every  $g_{ij}$  is continuously differentiable ([23, pp. 218, 253]). Hence  $q$  given by (4.4) is also continuously differentiable. Its partial derivative  $\partial q(p)/\partial p_i$ , to be denoted by  $d_i(p)$ , is given by

$$(4.5) \quad d_i(p) = \frac{\partial q(p)}{\partial p_i} = \sum_{j \in \mathfrak{D}(i)} \nabla g_{ij}(p_i - p_j) - \sum_{j \in \mathfrak{U}(i)} \nabla g_{ji}(p_j - p_i) - s_i.$$

Given a price vector  $p \in \mathfrak{R}^n$ , we consider an iteration whereby the dual objective function  $q$  is minimized with respect to the  $i$ th coordinate  $p_i$ , while the remaining coordinates are held fixed. In view of the convexity and the differentiability of  $q$ , this is equivalent to solving the equation  $d_i(p_1, \dots, p_{i-1}, \theta, p_{i+1}, \dots, p_n) = 0$  with respect to the scalar  $\theta$ . This equation can have several solutions and we will consider a mapping which chooses the solution that is nearest to the original price  $p_i$ . Accordingly, we define a function  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  whose  $i$ th coordinate is given by

$$(4.6) \quad h_i(p) = \operatorname{argmin} \{|\theta - p_i| \mid d_i(p_1, \dots, p_{i-1}, \theta, p_{i+1}, \dots, p_n) = 0\}.$$

We will show later in Lemma 4.1 that the set in (4.6) is nonempty and the minimum in (4.6) is attained, so that  $h$  is well defined. Notice that  $h(p) = p$  if and only if  $\partial q(p)/\partial p_i = d_i(p) = 0$  for every  $i$ . It follows that  $P^*$  is the set of fixed points of  $h$ .

Since  $q$  is convex, the set  $P^*$  is convex ( $P^*$  is also nonempty by assumption). Also from Proposition 2.3 in [6] we have that, for any  $p \in \mathfrak{R}^n$  and any  $p^* \in P^*$ ,

$$\min_{j \in \mathcal{N}} \{p_j - p_j^*\} \leq h_i(p) - p_i^* \leq \max_{j \in \mathcal{N}} \{p_j - p_j^*\}, \quad \forall i \in \mathcal{N},$$

and hence  $h$  has the pseudo-nonexpansive property

$$\|h(p) - p^*\| \leq \|p - p^*\|.$$

Furthermore, by using Proposition 1 in [5] and an argument analogous to the proof of Proposition 2.5 in Chapter 7.2 of [7], we can show that the mapping  $h$  is continuous. Therefore,  $h$  satisfies parts (a)-(c) of Assumption B. We show below that  $h$  is well defined and also satisfies part (d) of Assumption B.

LEMMA 4.1. *The mapping  $h$  is well defined and satisfies Assumption B(d).*

*Proof.* We start by mentioning certain facts that will be freely used in the course of the proof.

(a) For any  $(i, j) \in \mathcal{A}$ , the function  $\nabla g_{ij}$  is nondecreasing. (This is because  $g_{ij}$  is convex.)

(b)  $d_i: \mathfrak{R}^n \rightarrow \mathfrak{R}$  is a nondecreasing function of the  $i$ th coordinate of its argument when the other coordinates are held fixed. (This is because the dual functional  $q$  is convex and  $d_i = \partial q / \partial p_i$ .)

(c) A vector  $p^* \in \mathfrak{R}^n$  belongs to  $P^*$  if and only if, for every arc  $(i, j)$ , we have  $\nabla g_{ij}(p_i^* - p_j^*) = f_{ij}^*$ . (This is a direct consequence of the Network Equilibrium Theorem in [24, p. 349].)

We first show that  $h$  is well defined. Fix any  $p \in \mathfrak{R}^n$  and any  $i$ . We claim that there exists  $\theta_1$  such that  $d_i(p + \theta_1 e^i) \leq 0$ , where  $e^i$  denotes the  $i$ th coordinate vector in  $\mathfrak{R}^n$ . To see this, let  $p^*$  be any element of  $P^*$  and let  $\theta_1$  be any scalar sufficiently large so that

$$p_i - p_j + \theta_1 \geq p_i^* - p_j^*, \quad \forall j \in \mathcal{D}(i),$$

$$p_j - p_i - \theta_1 \leq p_j^* - p_i^*, \quad \forall j \in \mathcal{U}(i).$$

Since  $\nabla g_{kl}$  is nondecreasing for all  $(k, l) \in \mathcal{A}$ , this implies that

$$\nabla g_{ij}(p_i - p_j + \theta_1) \geq \nabla g_{ij}(p_i^* - p_j^*) = f_{ij}^*, \quad \forall j \in \mathcal{D}(i),$$

$$\nabla g_{ji}(p_j - p_i - \theta_1) \leq \nabla g_{ji}(p_j^* - p_i^*) = f_{ji}^*, \quad \forall j \in \mathcal{U}(i).$$

Upon summing the above inequalities, we obtain that

$$\begin{aligned} d_i(p + \theta_1 e^i) &= \sum_{j \in \mathcal{D}(i)} \nabla g_{ij}(p_i - p_j + \theta_1) - \sum_{j \in \mathcal{U}(i)} \nabla g_{ji}(p_j - p_i - \theta_1) - s_i \\ &\geq \sum_{j \in \mathcal{D}(i)} f_{ij}^* - \sum_{j \in \mathcal{U}(i)} f_{ji}^* - s_i \\ &= 0, \end{aligned}$$

where the last equality follows because the flows  $f_{ij}^*$  and  $f_{ji}^*$  must satisfy the flow conservation equation (4.2). Similarly, we can show that there exists  $\theta_2$  such that  $d_i(p + \theta_2 e^i) \leq 0$ . Since  $d_i(p + \theta e^i)$  is a continuous function of  $\theta$ , this implies that there exists some  $\theta$  between  $\theta_1$  and  $\theta_2$  such that  $d_i(p + \theta e^i) = 0$ . Therefore the set in (4.6) is nonempty. Since this set is also convex (due to the convexity of  $q$ ) and closed (due to the continuity of  $d_i$ ), the minimum in (4.6) is attained. Hence  $h$  is well defined.

Now we show that  $h$  satisfies Assumption B(d). We will argue by contradiction. Suppose that  $h$  does not satisfy Assumption B(d). Then for some  $p \notin P^*$  and  $p^* \notin P^*$  such that  $\|p - p^*\| = \rho(p) > 0$  there exists, for every  $i \in I(p; p^*)$ , a vector  $p^i \in U(p; p^*)$  such that  $h_i(p^i) = p_i^i$ . ( $\rho(p)$  denotes the maximum norm distance of  $p$  from  $P^*$ .) Let  $\beta = \rho(p)$ ,  $J = I(p; p^*)$ ,  $\varepsilon = \beta - \max\{\|p_i^k - p_i^*\| \mid i \in J, k \in J\}$ , and

$$J^- = \{i \mid p_i - p_i^* = -\beta\},$$

$$J^+ = \{i \mid p_i - p_i^* = \beta\}.$$

Then  $\varepsilon > 0$ ,  $J = J^- \cup J^+$  and, for all  $i \in J$ ,

$$(4.7) \quad p_j^* - \beta + \varepsilon \leq p_j^i \leq p_j^* + \beta - \varepsilon, \quad \forall j \notin J,$$

$$(4.8a) \quad p_j^i = p_j^* - \beta, \quad \forall j \in J^-,$$

$$(4.8b) \quad p_j^i = p_j^* + \beta, \quad \forall j \in J^+.$$

Fix any  $i \in J^-$ . The relations (4.7), (4.8a) imply that

$$p_i^i - p_j^i \cong (p_i^* - \beta) - (p_j^* - \beta) = p_i^* - p_j^*, \quad \forall j \in \mathcal{D}(i),$$

$$p_j^i - p_i^i \cong (p_j^* - \beta) - (p_i^* - \beta) = p_j^* - p_i^*, \quad \forall j \in \mathcal{U}(i),$$

and, since  $\nabla g_{kl}$  is nondecreasing for all  $(k, l) \in \mathcal{A}$ ,

$$(4.9a) \quad \nabla g_{ij}(p_i^i - p_j^i) \cong \nabla g_{ij}(p_i^* - p_j^*) = f_{ij}^*, \quad \forall j \in \mathcal{D}(i),$$

$$(4.9b) \quad \nabla g_{ji}(p_j^i - p_i^i) \cong \nabla g_{ji}(p_j^* - p_i^*) = f_{ji}^*, \quad \forall j \in \mathcal{U}(i).$$

Since  $i \in J^-$ , we have  $h_i(p^i) = p_i^i$  or, equivalently,  $d_i(p^i) = 0$ . Then (4.5) and (4.9a)-(4.9b) imply that

$$\begin{aligned} 0 &= d_i(p^i) \\ &= \sum_{j \in \mathcal{D}(i)} \nabla g_{ij}(p_i^i - p_j^i) - \sum_{j \in \mathcal{U}(i)} \nabla g_{ji}(p_j^i - p_i^i) - s_i \\ &\leq \sum_{j \in \mathcal{D}(i)} f_{ij}^* - \sum_{j \in \mathcal{U}(i)} f_{ji}^* - s_i \\ &= 0, \end{aligned}$$

where the last equality follows because the flows  $f_{ij}^*$  and  $f_{ji}^*$  must satisfy the flow conservation equation (4.2). It follows that the inequalities in (4.9a)-(4.9b) are actually equalities and

$$(4.10a) \quad \nabla g_{ij}(p_i^i - p_j^i) = f_{ij}^*, \quad \forall j \in \mathcal{D}(i),$$

$$(4.10b) \quad \nabla g_{ji}(p_j^i - p_i^i) = f_{ji}^*, \quad \forall j \in \mathcal{U}(i).$$

Since the choice of  $i \in J^-$  was arbitrary, (4.10a)-(4.10b) hold for all  $i \in J^-$ . By an analogous argument (using (4.8b) in place of (4.8a)) we can show that (4.10a)-(4.10b) hold for all  $i \in J^+$  as well.

Let  $\pi \in \mathfrak{R}^n$  be the vector whose  $i$ th component is

$$(4.11) \quad \pi_i = \begin{cases} p_i^* + \varepsilon & \text{if } i \in J^+, \\ p_i^* - \varepsilon & \text{if } i \in J^-, \\ p_i^* & \text{if } i \notin J. \end{cases}$$

We claim that

$$(4.12) \quad \nabla g_{ij}(\pi_i - \pi_j) = f_{ij}^*, \quad \forall (i, j) \in \mathcal{A}.$$

To see this, we first note from the definition of  $\pi$  (cf. (4.11)) that

$$\pi_i - \pi_j = p_i^* - p_j^*, \quad \text{if } i \notin J, j \notin J \quad \text{or if } i \in J^+, j \in J^+ \quad \text{or if } i \in J^-, j \in J^-.$$

Also, from (4.7), (4.8a)-(4.8b), (4.11) and the fact  $\varepsilon \leq \beta$  we have that

$$p_i^i - p_j^i = (p_i^* + \beta) - (p_j^* - \beta) \cong \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \in J^+, j \in J^-,$$

$$p_i^i - p_j^i = (p_i^* - \beta) - (p_j^* + \beta) \cong \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \in J^-, j \in J^+,$$

$$p_i^i - p_j^i \cong (p_i^* + \beta) - (p_j^* + \beta - \varepsilon) = \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \in J^+, j \notin J,$$

$$p_i^i - p_j^i \cong (p_i^* - \beta) - (p_j^* - \beta + \varepsilon) = \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \in J^-, j \notin J,$$

$$p_i^i - p_j^i \cong (p_i^* + \beta - \varepsilon) - (p_j^* + \beta) = \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \notin J, j \in J^+,$$

$$p_i^i - p_j^i \cong (p_i^* - \beta + \varepsilon) - (p_j^* - \beta) = \pi_i - \pi_j \cong p_i^* - p_j^*, \quad \text{if } i \notin J, j \in J^-.$$

Consider any  $(i, j) \in \mathcal{A}$ . The preceding inequalities show that  $\pi_i - \pi_j$  is always between  $p_i^i - p_j^j$  and  $p_i^* - p_j^*$ . The monotonicity of  $\nabla g_{ij}$  and the equalities  $\nabla g_{ij}(p_i^* - p_j^*) = f_{ij}^* = \nabla g_{ij}(p_i^i - p_j^j)$  (cf. (4.10a)-(4.10b)) imply that  $\nabla g_{ij}(\pi_i - \pi_j) = f_{ij}^*$ . This completes the proof of (4.12).

Equation (4.12) implies that  $\pi \in P^*$ . Since (cf. (4.11) and the definitions of  $J^-$  and  $J^+$ )  $\|p - \pi\| < \|p - p^*\|$ , this contradicts the hypothesis that  $\rho(p) = \|p - p^*\|$ .  $\square$

Since  $h$  has been shown to satisfy Assumption B, we conclude from Lemma 2.1 and Proposition 2.1 that the partially asynchronous iteration

$$p := (1 - \gamma)p + \gamma h(p)$$

(with  $0 < \gamma < 1$ ) converges to an optimal price vector  $p^*$ . The optimal flows are obtained as a byproduct, using the relation  $\nabla g_{ij}(p_i^* - p_j^*) = f_{ij}^*$ . Notice that the iteration for each coordinate  $p_i$  consists of minimization along the  $i$ th coordinate direction (to obtain  $h_i(p)$ ) followed by the use of the relaxation parameter  $\gamma$  to obtain the new value  $(1 - \gamma)p_i + \gamma h_i(p)$ . As a special case, we have that the synchronous Jacobi algorithm

$$p(t+1) = (1 - \gamma)p(t) + \gamma h(p(t))$$

is also convergent, which is a new result.

A related result can be found in [5] where totally asynchronous convergence is established even if  $\gamma = 1$ , provided that a particular coordinate of  $p$  is never iterated upon and that when this coordinate is fixed, the optimal price vector is unique. An experimental comparison of the two methods will be presented in § 8. We remark that the results in this section also extend to the case where each arc has a gain of either +1 or -1 (i.e., each  $f_{ij}$  term in (4.2) is multiplied by either +1 or -1).

**5. Agreement and Markov chain algorithms.** In this section we consider two problems: a problem of agreement and the computation of the invariant distribution of a Markov chain. These problems are the only ones for which partially asynchronous algorithms that converge for every value of the asynchronism bound  $B$  of Assumption A are available [20], [27], [29] (in fact, these algorithms have been shown to converge at a geometric rate). We show that these results can also be obtained by applying our general convergence theorem (Proposition 2.1).

**5.1. The agreement algorithm.** We consider here a set of  $n$  processors, numbered from 1 to  $n$ , that try to reach agreement on a common value by exchanging tentative values and forming convex combinations of their own values with the values received from other processors. This algorithm has been used in [28]-[29] in the context of asynchronous stochastic gradient methods with the purpose of averaging noisy measurements of the same variable by different processors.

We now formally describe the agreement algorithm. Each processor  $i$  has a set of nonnegative coefficients  $\{a_{i1}, \dots, a_{in}\}$  satisfying  $a_{ii} > 0$ ,  $\sum_j a_{ij} = 1$ , and at time  $t$  it possesses an estimate  $x_i(t)$  which is updated according to (cf. (1.1))

$$(5.1a) \quad x_i(t+1) = \begin{cases} \sum_{j=1}^n a_{ij} x_j(\tau_{ij}(t)) & \text{if } t \in \mathcal{T}_i, \\ x_i(t) & \text{otherwise.} \end{cases}$$

$$(5.1b) \quad x_i(1 - B) = \dots = x_i(0) = \bar{x}_i,$$

where  $\mathcal{T}_i$  and  $\tau_{ij}(t)$  are as in § 1 and  $\bar{x}_i$  is the initial value of processor  $i$ . Let  $A$  be the  $n \times n$  matrix whose  $(i, j)$ th entry is  $a_{ij}$  and let  $\gamma \in (0, 1)$  be such that  $0 < \gamma \leq \min\{a_{11}, \dots, a_{nn}\}$ . By using the results from §§ 1 to 3 we obtain the following.

**PROPOSITION 5.1.** *If  $A$  is irreducible and Assumption A holds, then  $\{x_i(t)\} \rightarrow y$  for all  $i$ , where  $y$  is some scalar between  $\min_i \{\bar{x}_i\}$  and  $\max_i \{\bar{x}_i\}$ .*

*Proof.* It can be seen that (5.1a) is a special case of (1.1) with  $f(x) = Ax$ . Let

$$D = (A - \gamma I)/(1 - \gamma).$$

Then

$$(5.2) \quad A = \gamma I + (1 - \gamma)D,$$

and  $D = [d_{ij}]$  can be seen to satisfy  $\sum_j |d_{ij}| \leq 1$ . Moreover, since  $A$  is irreducible, so is  $D$ . Hence the function  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  defined by  $h(x) = Dx$  satisfies Assumption D in § 3. Since  $h$  has a fixed point (the zero vector), this, together with Proposition 3.1 and Lemma 2.4, implies that  $h$  satisfies Assumption B. Since (cf. (5.2))  $f(x) = \gamma x + (1 - \gamma)h(x)$ , this, together with Lemma 2.1, shows that  $f$  satisfies Assumption C. Then by Proposition 2.1, the sequence  $\{x(t)\}$  generated by (5.1a)–(5.1b) converges to some point  $x^\infty$  satisfying  $Ax^\infty = x^\infty$ . Since  $A$  is irreducible and stochastic,  $x^\infty$  must be of the form  $(y, \dots, y)$  for some  $y \in \mathfrak{R}$ . It can be seen from (5.1b) that, for  $r \in \{1 - B, \dots, 0\}$ ,

$$(5.3) \quad x_i(r) \leq \max_j \{\bar{x}_j\}, \quad \forall i.$$

Suppose that (5.3) holds for all  $r \in \{1 - B, \dots, t\}$ , for some  $t \geq 0$ . Then by (5.1a) and the property of the  $a_{ij}$ 's,

$$\begin{aligned} x_i(t+1) &= \sum_j a_{ij} x_j(\tau_{ij}(t)) \\ &\leq \sum_j a_{ij} \max_j \{\bar{x}_j\} \\ &= \max_j \{\bar{x}_j\}, \end{aligned}$$

for all  $i$  such that  $t \in \mathcal{T}_i$ , and  $x_i(t+1) = x_i(t) \leq \max_j \{\bar{x}_j\}$  for all other  $i$ . Hence, by induction, (5.3) holds for all  $r \in \{1 - B, 2 - B, \dots\}$ . Since  $x_i(r) \rightarrow y$  for each  $i$ , this implies that  $y \leq \max_j \{\bar{x}_j\}$ . A symmetrical argument shows  $y \geq \min_j \{\bar{x}_j\}$ .  $\square$

It can be shown [7], [29] that Proposition 5.1 remains valid if  $a_{ii}$  is positive for at least one (but not all)  $i$  and, furthermore, convergence takes place at the rate of a geometric progression. The proof, however, is more complex. Similar results can be found in [29] for more general versions of the agreement algorithm.

**5.2. Invariant distribution of Markov chains.** Let  $P$  be an irreducible stochastic matrix of dimension  $n \times n$ . We denote by  $p_{ij}$  the  $(i, j)$ th entry of  $P$  and we assume that  $p_{ii} > 0$  for all  $i$ . We wish to compute a row vector  $\pi^* = (\pi_1^*, \dots, \pi_n^*)$  of invariant probabilities for the corresponding Markov chain, i.e.,  $\pi_i^* \geq 0$ ,  $\sum_i \pi_i^* = 1$ ,  $\pi^* = \pi^* P$ . (We actually have  $\pi_i^* > 0$ , for all  $i$ , due to the irreducibility of  $P$  [14].) As in § 5.1, suppose that we have a network of  $n$  processors and that the  $i$ th processor generates a sequence of estimates  $\{\pi_i(t)\}$  using the following partially asynchronous version of the classical serial algorithm  $\pi := \pi P$  (cf. (5.1a)–(5.1b)):

$$(5.4) \quad \pi_i(t+1) = \begin{cases} \sum_{j=1}^n p_{ji} \pi_j(\tau_{ij}(t)) & \text{if } t \in \mathcal{T}_i, \\ \pi_i(t) & \text{otherwise.} \end{cases}$$

$$\pi_i(1 - B) = \dots = \pi_i(0),$$

where  $\mathcal{T}_i$  and  $\tau_{ij}(t)$  are as in § 1 and  $\pi_i(0)$  is any positive scalar. This asynchronous algorithm was introduced in [20], where geometric convergence was established. We show below that convergence also follows from our general results.

PROPOSITION 5.2. *If Assumption A holds, then there exists a positive number  $c$  such that  $\pi(t) \rightarrow c\pi^*$ .*

*Proof.* We will show that (5.4) is a special case of (5.1a). Let

$$(5.5) \quad x_i(t) = \pi_i(t)/\pi_i^*, \quad a_{ij} = \pi_j^* p_{ji}/\pi_i^*.$$

Then the matrix  $A = [a_{ij}]$  is nonnegative and irreducible, has positive diagonal entries, and

$$\begin{aligned} \sum_j a_{ij} &= \sum_j \pi_j^* p_{ji}/\pi_i^* \\ &= \pi_i^*/\pi_i^* \\ &= 1, \end{aligned}$$

where the second equality follows from  $\pi^* = \pi^*P$ . Furthermore, it can be seen from (5.4) and (5.5) that  $x_i(t)$  evolves according to the iteration (5.1a). Therefore, by Proposition 5.1 and the initial positivity conditions,  $\{x_i(t)\} \leftrightarrow c$  for all  $i$ , where  $c$  is some positive scalar. It follows from (5.5) that  $\pi_i(t) \rightarrow c\pi_i^*$  for all  $i$ .  $\square$

Upon obtaining  $c\pi^*$ , the desired solution  $\pi^*$  can be recovered by normalizing  $c\pi^*$ .

**6. Neural networks.** Consider a connected, directed network with node set  $\mathcal{N} = \{1, \dots, n\}$  and arc set  $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ . Let us, for each  $i \in \mathcal{N}$ , denote by  $\mathcal{U}(i)$  the set  $\{j | (j, i) \in \mathcal{A}\}$  of upstream neighbors of  $i$ . Let  $\sigma_1, \dots, \sigma_n$  be a set of given scalars and let  $\{\lambda_{ij}\}_{(i,j) \in \mathcal{A}}$  be a set of nonzero scalars satisfying  $\sum_{j \in \mathcal{U}(i)} |\lambda_{ij}| \leq 1$  for all  $i$ . We wish to find scalars  $x_1, \dots, x_n$  such that

$$(6.1) \quad x_i = \phi_i \left( \sum_{j \in \mathcal{U}(i)} \lambda_{ij} x_j + \sigma_i \right), \quad \forall i,$$

where  $\phi_i: \mathfrak{R} \rightarrow \mathfrak{R}$  is a continuous nondecreasing function satisfying

$$(6.2) \quad \lim_{\xi \rightarrow -\infty} \phi_i(\xi) = -1, \quad \lim_{\xi \rightarrow +\infty} \phi_i(\xi) = 1,$$

(see Fig. 6.1). Notice that the function  $\phi_i$  maps the box  $[-1, 1]^n$  into itself and, by Brouwer's fixed point theorem ([11, p. 17]), the system (6.1) is guaranteed to have a solution.

If we think of each node  $i$  as a neuron, (6.1) and (6.2) imply that this neuron is turned on (i.e.,  $x_i \approx 1$ ) if the majority of its inputs are also turned on. Thus,  $x_i$  gives the state ("on" or "off") of the  $i$ th neuron for a given set of connections (specified by  $\mathcal{A}$ ) and a given external excitation (specified by  $\sigma_i$ ) (see Fig. 6.2.). Indeed, (6.1) and (6.2) describe a class of *neural networks* that have been applied to solving a number of problems in combinatorial optimization, pattern recognition, and artificial intelligence [15]-[16], [19], [25].

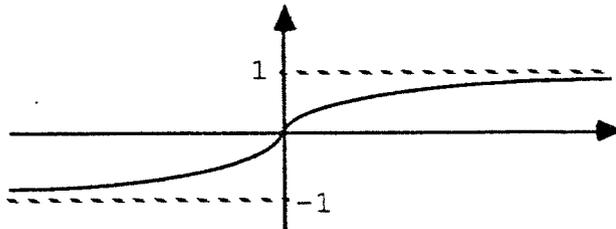


FIG. 6.1. The function  $\phi_i$ .

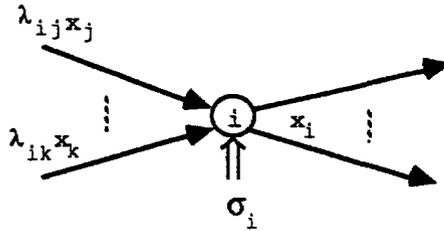


FIG. 6.2

Let  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  be the function whose  $i$ th component is

$$(6.3) \quad f_i(x) = \phi_i \left( \sum_{j \in \mathcal{Q}(i)} \lambda_{ij} x_j + \sigma_i \right), \quad \forall i.$$

Then solving (6.1) is equivalent to finding a fixed point of  $f$ . In what follows, we consider a special form for  $\phi_i$  and show that it gives rise, in a natural way, to a nonexpansive function  $f$  that satisfies Assumptions B' and C of § 2. To the best of our knowledge, asynchronous convergence of neural networks has not been explored before. In some sense, asynchronous neural networks are quite natural since biological neural connections may experience long propagation delay [25].

Let  $\phi_i^+$  denote the *right derivative* of  $\phi_i$ , i.e.,

$$\phi_i^+(\xi) = \lim_{\varepsilon \downarrow 0} (\phi_i(\xi + \varepsilon) - \phi_i(\xi)) / \varepsilon, \quad \forall \xi \in \mathfrak{R}.$$

The following result shows that, if  $\phi_i^+$  is sufficiently small for all  $i$ , then  $f$  given by (6.3) satisfies Assumption B'.

**PROPOSITION 6.1.** *If  $\mathcal{G}$  is strongly connected and each  $\phi_i$  is continuous, satisfies (6.2) and*

$$(6.4) \quad 0 \leq \phi_i^+(\xi) \leq 1, \quad \forall \xi \in \mathfrak{R},$$

then  $f$  given by (6.3) satisfies Assumption B'.

*Proof.* We have seen earlier that  $f$  has a fixed point. Since each  $\phi_i$  is continuous,  $f$  is also continuous. Now we will show that  $f$  is nonexpansive. Fix any  $i \in \mathcal{N}$ . Since (cf. (6.4)) the slope of  $\phi_i$  is bounded inside the interval  $[0, 1]$ , we have

$$|\phi_i(b) - \phi_i(a)| \leq |b - a|, \quad \forall a \in \mathfrak{R}, b \in \mathfrak{R}.$$

Hence, for any  $x \in \mathfrak{R}^n$  and  $y \in \mathfrak{R}^n$ ,

$$(6.5) \quad \begin{aligned} |f_i(y) - f_i(x)| &= \left| \phi_i \left( \sum_{j \in \mathcal{Q}(i)} \lambda_{ij} y_j + \sigma_i \right) - \phi_i \left( \sum_{j \in \mathcal{Q}(i)} \lambda_{ij} x_j + \sigma_i \right) \right| \\ &\leq \left| \sum_{j \in \mathcal{Q}(i)} \lambda_{ij} (y_j - x_j) \right| \\ &\leq \sum_{j \in \mathcal{Q}(i)} |\lambda_{ij}| |y_j - x_j|. \end{aligned}$$

Since  $\sum_{j \in \mathcal{Q}(i)} |\lambda_{ij}| \leq 1$ , (6.5) implies that

$$|f_i(y) - f_i(x)| \leq \|x - y\|.$$

Since the choice of  $i$  was arbitrary, this in turn implies that

$$\|f(x) - f(y)\| \leq \|x - y\|, \quad \forall x \in \mathfrak{R}^n, y \in \mathfrak{R}^n.$$

Therefore  $f$  is nonexpansive.

It remains to show that  $f$  satisfies Assumption B'(d). Suppose the contrary. Then for some  $x \notin X^*$  and some  $x^* \in X^*$ , where  $X^*$  is the set of fixed points of  $f$ , there exists, for every  $i \in I(x; x^*)$ , an  $x' \in U(x; x^*)$  such that

$$x' \notin X^* \quad \text{and} \quad f_i(x') = x'_i.$$

Let  $J = I(x; x^*)$  ( $J \neq \mathcal{A}$  since  $x' \notin X^*$  for all  $i \in J$ ) and  $\beta = \|x - x^*\|$ . Fix any  $i \in J$ . By (6.5) and the fact  $x^* = f(x^*)$ , we obtain that

$$|x'_i - x^*_i| = |f_i(x') - f_i(x^*)| \leq \sum_{j \in \mathcal{U}(i)} |\lambda_{ij}| |x'_j - x^*_j|.$$

Hence

$$\begin{aligned} \beta &\leq \sum_{j \in \mathcal{U}(i)} |\lambda_{ij}| |x'_j - x^*_j| \\ &= \sum_{j \in \mathcal{U}(i)} |\lambda_{ij}| \beta + \sum_{j \in \mathcal{U}(i), j \in J} |\lambda_{ij}| (|x'_j - x^*_j| - \beta) \\ &\leq \beta + \sum_{j \in \mathcal{U}(i), j \in J} |\lambda_{ij}| (|x'_j - x^*_j| - \beta). \end{aligned}$$

Since  $|x'_j - x^*_j| < \beta$  and  $\lambda_{ij} \neq 0$  for all  $j \in \mathcal{U}(i), j \in J$ , this implies that  $\mathcal{U}(i) \subseteq J$ . Since the choice of  $i \in J$  was arbitrary, it follows that  $\mathcal{U}(i) \subseteq J$  for all  $i \in J$ . Hence  $\mathcal{G}$  is not strongly connected, a contradiction of our hypothesis.  $\square$

It follows from Lemmas 2.1, 2.4 and Propositions 2.1, 6.1 that the asynchronous iteration

$$x_i := (1 - \gamma)x_i + \gamma \phi_i \left( \sum_{j \in \mathcal{U}(i)} \lambda_{ij} x_j + \sigma_i \right),$$

(with  $0 < \gamma < 1$ ) converges. Two examples of  $\phi_i$  that satisfy the hypothesis of Proposition 6.1 are

$$\phi_i(\xi) = 2(1 + e^{-2\xi})^{-1} - 1,$$

and

$$\phi_i(\xi) = \max \{-1, \min \{1, \xi\}\}.$$

Let us briefly discuss an alternative form for the function  $\phi_i$ . If we assume that each  $\phi_i$  is continuously differentiable and its derivative  $\nabla \phi_i$  satisfies  $0 < \nabla \phi_i(\xi) < 1$  for all  $\xi \in \mathfrak{R}$ , then it can be shown that the restriction of the function  $f$  on a compact set is a contraction. In that case, the asynchronous neural iteration

$$x_i := \phi_i \left( \sum_{j \in \mathcal{U}(i)} \lambda_{ij} x_j + \sigma_i \right)$$

can be shown to converge even under the total asynchronism assumption

$$\lim_{t \rightarrow +\infty} \tau_{ij}(t) = +\infty, \quad \forall i, \quad \forall j$$

(cf. [7, Chap. 6.2, Prop. 2.1]).

**7. Least element of weakly diagonally dominant, Leontief systems.** Let  $A = [a_{kj}]$  be a given  $m \times n$  matrix (with  $m \geq n$ ) and  $b = (b_1, \dots, b_m)$  be an element of  $\mathfrak{R}^m$ . We make the following assumption.

*Assumption F.*

(a) Each row of  $A$  has exactly one positive entry and the index set

$$I(i) = \{k | a_{ki} > 0\}$$

is nonempty for all  $i$  (i.e., every column has at least one positive entry).

(b)  $\sum_j a_{kj} \geq 0$ , for all  $k$ .

(c) For any  $(k_1, \dots, k_n) \in I(1) \times \dots \times I(n)$ , the  $n \times n$  matrix  $[a_{k_i j}]$  is irreducible.

Since  $a_{ki} > 0$  for all  $k \in I(i)$ , we will, by dividing the  $k$ th constraint by  $a_{ki}$  if necessary, assume that  $a_{ki} = 1$  for all  $k \in I(i)$ , in which case parts (a) and (b) of Assumption F are equivalent to

$$(7.1) \quad a_{ki} = 1, \quad -\sum_{j \neq i} a_{kj} \leq 1 \quad \text{and} \quad a_{kj} \leq 0, \quad \forall j \neq i,$$

for all  $k \in I(i)$  and all  $i$ .

Let  $X$  be the polyhedral set

$$(7.2) \quad X = \{x \in \mathfrak{R}^n \mid Ax \geq b\}.$$

We wish to find an element  $\eta$  of  $X$  satisfying

$$x \geq \eta, \quad \forall x \in X$$

(such an element is called the *least element* of  $X$  in [10] and [13]). Notice that if a least element exists, then it is unique. Let  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  be the function whose  $i$ th component is

$$(7.3) \quad h_i(x) = \max_{k \in I(i)} \left\{ b_k - \sum_{j \neq i} a_{kj} x_j \right\}.$$

It is shown in [10] that  $X$  has a least element for all  $b$  such that  $X$  is nonempty if and only if  $A^T$  is *Leontief* (a matrix  $E$  is Leontief if each column of  $E$  has at most one positive entry and there exists  $y \geq 0$  such that  $Ey > 0$  componentwise). The following lemma sharpens this result by giving a necessary and sufficient condition for  $X$  to have a least element that is simpler to verify. It also relates the least element of  $X$  to the fixed points of  $h$ .

LEMMA 7.1. *Suppose that  $X \neq \emptyset$  and that Assumption F holds. Then,*

(a)  *$X$  has no least element if and only if*

$$(7.4) \quad \sum_j a_{kj} = 0, \quad \forall k.$$

(b) *If  $\eta$  is a least element of  $X$ , then it is a fixed point of  $h$ .*

*Proof.* We first prove (a). Suppose that (7.4) holds and let  $e \in \mathfrak{R}^n$  be the vector with all coordinates equal to 1. Equation (7.4) says that  $Ae = 0$ . Thus, if  $x$  is an element of  $X$ , then  $x - \lambda e \in X$ , for all positive scalars  $\lambda$ . It follows that  $X$  cannot have a least element. Now suppose that (7.4) does not hold. We first show that  $X$  is bounded from below (i.e., there exists some  $a \in \mathfrak{R}^n$  such that  $x \geq a$  componentwise for all  $x \in X$ ). If this were not so, then there would exist some  $v \in \mathfrak{R}^n$  and some  $x \in X$  such that  $v_i < 0$  for some  $i$  and  $x + \lambda v \in X$  for all positive scalars  $\lambda$ . The latter implies that  $A(x + \lambda v) \geq b$  for all  $\lambda > 0$  and hence  $Av \geq 0$ . Fix any scalars  $(k_1, \dots, k_n) \in I(1) \times \dots \times I(n)$  and consider an  $i$  such that  $v_i = \min_j \{v_j\}$ . Then (cf.  $Av \geq 0$ )

$$0 \leq \sum_j a_{k_i j} v_j = \left( \sum_j a_{k_i j} \right) v_i + \sum_{j \neq i} a_{k_i j} (v_j - v_i).$$

Since  $v_i < 0$  and  $v_j - v_i \geq 0$  for all  $j \neq i$ , this, together with the facts (cf. (7.1))  $\sum_j a_{k_i j} \geq 0$  and  $a_{k_i j} \leq 0$  for all  $j \neq i$ , implies that  $\sum_j a_{k_i j} = 0$  and  $v_i = v_j$  for all  $j \neq i$  such that  $a_{k_i j} \neq 0$ . By Assumption F(c), there exists  $j \neq i$  such that  $a_{k_i j} \neq 0$ . We then repeat the above argument with  $j$  in place of  $i$ . In this way, we eventually obtain that  $v_1 = \dots = v_n$  and  $\sum_j a_{k_i j} = 0$  for all  $i$ . Since our choice of  $(k_1, \dots, k_n) \in I(1) \times \dots \times I(n)$  was arbitrary, (7.4) holds—contradicting our hypothesis. Hence  $X$  is bounded from below. Using

(7.1), it is easily verified that if  $x'$  and  $x''$  are two elements of  $X$ , then the  $n$ -vector  $x$  whose  $i$ th component is  $\min \{x'_i, x''_i\}$  is also an element of  $X$ . Since  $X$  is closed and bounded from below,  $X$  has a least element.

We next prove (b). Since  $\eta \in X$ , we have (cf. (7.1), (7.2))

$$\sum_{j \neq i} a_{kj} \eta_j + \eta_i \geq b_k, \quad \forall k \in I(i), \quad \forall i.$$

Thus,

$$h_i(\eta) = \max_{k \in I(i)} \left\{ b_k - \sum_{j \neq i} a_{kj} \eta_j \right\} \leq \eta_i, \quad \forall i.$$

If  $\eta$  is not a fixed point of  $h$ , then the set  $I = \{i \mid h_i(\eta) < \eta_i\}$  is nonempty. Then we have

$$(7.5) \quad \sum_j a_{kj} \eta_j > b_k, \quad \forall k \in I(i), \quad \forall i \in I.$$

Consider the  $n$ -vector  $\tilde{\eta}$ , defined by  $\tilde{\eta}_i = \eta_i - \varepsilon$ , if  $i \in I$ , and  $\tilde{\eta}_i = \eta_i$ , otherwise. For  $\varepsilon$  positive and small enough, the inequalities (7.5) remain valid. On the other hand, for all  $i \notin I$  and all  $k \in I(i)$  we have

$$\sum_j a_{kj} \tilde{\eta}_j = \sum_{j \notin I} a_{kj} \eta_j + \sum_{j \in I} a_{kj} (\eta_j - \varepsilon) \geq \sum_j a_{kj} \eta_j \geq b_k,$$

where we used the property  $a_{kj} \leq 0$  for all  $j$  such that  $k \in I(j)$ . Thus,  $\tilde{\eta} \in X$ , contradicting the hypothesis that  $\eta$  is the least element of  $X$ .  $\square$

Let  $X^*$  denote the set of fixed points of  $h$ . Suppose that  $X^*$  is nonempty (Lemma 7.1 gives sufficient conditions for  $X^*$  to be nonempty). We will show that  $h$  satisfies Assumption B'. Since (cf. (7.3))  $h$  is continuous, it suffices to show that parts (c) and (d) of Assumption B' hold.

LEMMA 7.2.  $\|h(x) - h(y)\| \leq \|x - y\|$  for any  $x \in \mathfrak{R}^n$  and any  $y \in \mathfrak{R}^n$ .

*Proof.* Let  $z = h(x)$ ,  $w = h(y)$  and consider any  $i \in \{1, \dots, n\}$ . We will show that  $|z_i - w_i| \leq \|x - y\|$ , from which our claim follows. Since  $z_i = h_i(x)$  and  $w_i = h_i(y)$ , it follows from (7.3) that, for some  $k$  in  $I(i)$ ,

$$(7.6a) \quad \sum_{j \neq i} a_{kj} x_j + z_i \geq b_k,$$

$$(7.6b) \quad \sum_{j \neq i} a_{kj} y_j + w_i = b_k.$$

Subtracting (7.6b) from (7.6a), we obtain

$$\sum_{j \neq i} a_{kj} (x_j - y_j) + (z_i - w_i) \geq 0.$$

This together with (7.1) implies that

$$\begin{aligned} w_i - z_i &\leq \sum_{j \neq i} a_{kj} (x_j - y_j) \\ &\leq \sum_{j \neq i} |a_{kj}| \|x - y\| \\ &\leq \|x - y\|. \end{aligned}$$

The inequality  $z_i - w_i \leq \|x - y\|$  is obtained similarly.  $\square$

LEMMA 7.3.  $h$  satisfies Assumption B'(d).

*Proof.* Suppose the contrary. Then for some  $x \notin X^*$  and some  $x^* \in X^*$ , there exists, for every  $i \in I(x; x^*)$ , an  $x^i \in U(x; x^*)$  such that

$$x^i \notin X^* \quad \text{and} \quad h_i(x^i) = x^i.$$

Let  $J = I(x; x^*)$ ,  $J^- = \{i \mid x_i - x_i^* = -\beta\}$ ,  $J^+ = \{i \mid x_i - x_i^* = \beta\}$  and  $\beta = \|x - x^*\|$ . (We must have  $J \neq \{1, \dots, n\}$  because otherwise the set  $U(x; x^*)$  would be a singleton, implying that the vectors  $x^1, \dots, x^n$  are all equal, in which case each  $x^i$  is a fixed point of  $h$ , a contradiction.)

Fix any  $i \in J^-$ . By (7.3) and the hypothesis  $x^* = h(x^*)$ , there exists some  $k_i \in I(i)$  such that

$$\sum_J a_{k_i, j} x_j^* = b_{k_i}.$$

Since  $x_i^i = h_i(x^i)$ , we then have  $\sum_J a_{k_i, j} x_j^i \geq b_{k_i} = \sum_J a_{k_i, j} x_j^*$ , so

$$\sum_J a_{k_i, j} (x_j^i - x_j^*) \geq 0.$$

This implies (using (7.1) and the facts  $k_i \in I(i)$ ,  $i \in J^-$ ) that

$$\begin{aligned} 0 &\leq -\beta \sum_{j \in J^-} a_{k_i, j} + \beta \sum_{j \in J^+} a_{k_i, j} + \sum_{j \notin J} |a_{k_i, j}| |x_j^i - x_j^*| \\ &= -\beta \sum_{j \in J^-} a_{k_i, j} - \beta \sum_{j \in J^+} |a_{k_i, j}| + \sum_{j \notin J} |a_{k_i, j}| |x_j^i - x_j^*| \\ &= -\beta \left( 1 - \sum_{j \neq i} |a_{k_i, j}| \right) - 2\beta \sum_{j \in J^+} |a_{k_i, j}| + \sum_{j \notin J} |a_{k_i, j}| (|x_j^i - x_j^*| - \beta). \end{aligned}$$

Since  $|x_j^i - x_j^*| < \beta$  for all  $j \notin J$ , (7.1) implies that

$$(7.7) \quad \sum_{j \neq i} a_{k_i, j} = -1 \quad \text{and} \quad a_{k_i, j} = 0, \quad \forall j \notin J^-.$$

Since the choice of  $i$  was arbitrary, (7.7) holds for all  $i \in J^-$ . By an analogous argument, we also obtain that, for all  $i \in J^+$ ,

$$(7.8) \quad \sum_{j \neq i} a_{k_i, j} = -1 \quad \text{and} \quad a_{k_i, j} = 0, \quad \forall j \notin J^+,$$

where each  $k_i$  is a scalar in  $I(i)$  such that

$$\sum_J a_{k_i, j} x_j^i = b_{k_i}.$$

For each  $i \notin J$ , let  $k_i$  be any element of  $I(i)$ . Since  $J \neq \{1, \dots, n\}$ , (7.7) and (7.8) imply that the  $n \times n$  matrix  $[a_{k_i, j}]_{i, j}$  is *not* irreducible—a contradiction of Assumption F(c).  $\square$

We may now invoke Lemmas 2.1, 2.4 and Proposition 2.1 to establish that the partially asynchronous iteration

$$x := (1 - \gamma)x + \gamma h(x)$$

(with  $0 < \gamma < 1$ ) converges to a fixed point of  $h$ . Unfortunately, such a fixed point is not necessarily the least element of  $X$ . We have, however, the following characterization of such fixed points.

**LEMMA 7.4.** *If  $X$  has a least element  $\eta$ , then, for any fixed point  $x^*$  of  $h$ , there exists a nonnegative scalar  $\lambda$  such that  $x^* = \eta + (\lambda, \dots, \lambda)$ .*

*Proof.* Since  $x^*$  is a fixed point of  $h$ ,  $x^* \in X$ . Hence  $x^* \geq \eta$ . We then repeat the proof of Lemma 7.3, with  $J^- = \{1, \dots, n\}$  and  $x^i = \eta$  for all  $i$ . This yields that, for every  $i \in \{1, \dots, n\}$ , there exists some  $k_i \in I(i)$  such that  $x_i^* - \eta_i \leq \sum_{j \neq i} |a_{k_i j}| (x_j^* - \eta_j)$ . Since  $x^* - \eta \geq 0$ , Assumption F(c) and (7.1) imply that the  $x_i^* - \eta_i$ 's are equal.  $\square$

Lemma 7.4 states that, given a fixed point  $x^*$  of  $h$ , we can compute the least element of  $X$  by a simple line search along the direction  $(-1, \dots, -1)$  (the stepsize  $\lambda$  is the largest for which  $x^* - (\lambda, \dots, \lambda)$  is in  $X$ ). An example of  $X$  for which the corresponding  $h$  has multiple fixed points is

$$X = \{(x_1, x_2) \mid x_1 - x_2 \geq 0, x_1 - 0.5x_2 \geq -1, -x_1 + x_2 \geq 0\}.$$

Here  $h_1(x) = \max\{x_2, 0.5x_2 - 1\}$ ,  $h_2(x) = x_1$  and both  $(-1, -1)$  and  $(-2, -2)$  are fixed points of  $h$  (the least element of  $X$  is  $(-2, -2)$ ).

Let us remark that if the inequalities in Assumption F(b) are strict, then the mapping  $h$  is a contraction mapping (the same argument as in Lemma 7.2) and convergence under total asynchronism is obtained. We also remark that, if in the statement of Assumption F(c) we replace "For any" by the weaker "For some," then Lemmas 7.1 and 7.2 still hold, but Lemmas 7.3 and 7.4 do not. In fact, it can be shown that  $X^*$  is not necessarily convex in this case.

**8. Simulation for network flow problems.** In this section we study and compare, using simulation, the performance of synchronous and partially asynchronous algorithms for the network flow problem of § 4. We measure the following: (a) the effects of the stepsize  $\gamma$  (cf. Lemma 2.1), the problem size  $n$ , and the asynchrony measure  $B$  on the performance of partially asynchronous algorithms, (b) the efficiency of different partially asynchronous algorithms relative to each other and also relative to the corresponding synchronous algorithms.

In our study, we consider a special case of the network flow problem (4.1)–(4.2) where each cost function  $a_{ij}(\cdot)$  is a quadratic on  $[0, +\infty)$ , i.e.,

$$(8.1) \quad a_{ij}(f_{ij}) = \begin{cases} \alpha_{ij}|f_{ij}|^2 + \beta_{ij}f_{ij} & \text{if } f_{ij} \geq 0, \\ +\infty & \text{otherwise,} \end{cases}$$

where  $\alpha_{ij}$  is a given positive scalar and  $\beta_{ij}$  is a given scalar. This special case has many practical applications and has been studied extensively [6], [9], [12], [21], [31]. In what follows, we will denote by  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  the function given by (4.3), (4.5)–(4.6), and (8.1). All of the algorithms involved in our study are based on  $h$ .

**8.1. Test problem generation.** In our test, each  $\alpha_{ij}$  is randomly generated from the interval  $[1, 5]$  and each  $\beta_{ij}$  is randomly generated from the set  $\{1, 2, \dots, 100\}$ . The number of arcs is ten times the number of nodes and the average node supply is 1000, i.e.,  $|s_1| + \dots + |s_n| = 1000n$ . Half of the nodes are supply nodes and half of the nodes are demand nodes (we say a node  $i$  is a supply (demand) node if  $s_i > 0$  ( $s_i < 0$ )). The problems are generated using the linear cost network generator NETGEN [18], modified to generate quadratic cost coefficients as well.

**8.2. The main partially asynchronous algorithm.** The main focus of our study is the partially asynchronous algorithm described in § 4. This algorithm, called PASYN, generates a sequence  $\{x(t)\}$  using the partially asynchronous iteration (1.1) under Assumption A, where the algorithmic mapping  $f$  is given by

$$(8.2) \quad f(x) = (1 - \gamma)x + \gamma h(x).$$

In our simulation, the communication delays  $t - \tau_{ij}(t)$  are independently generated from a uniform distribution on the set  $\{0, 1, \dots, B-1\}$  and, for simplicity, we assume that  $\mathcal{T}_i = \{1, 2, \dots\}$  for all  $i$ . (This models a situation where the computation delay is negligible compared to the communication delay.) The components of  $x(1-B)$ ,  $x(2-B)$ ,  $\dots$ ,  $x(0)$  are independently generated from a uniform distribution over the interval  $[0, 10]$  (this is to reflect a lack of coordination among processors) and the algorithm terminates at time  $t$  if  $\max_{\tau, \tau' \in \{t-B, \dots, t\}} \|x(\tau) - x(\tau')\| \leq 0.001$ .

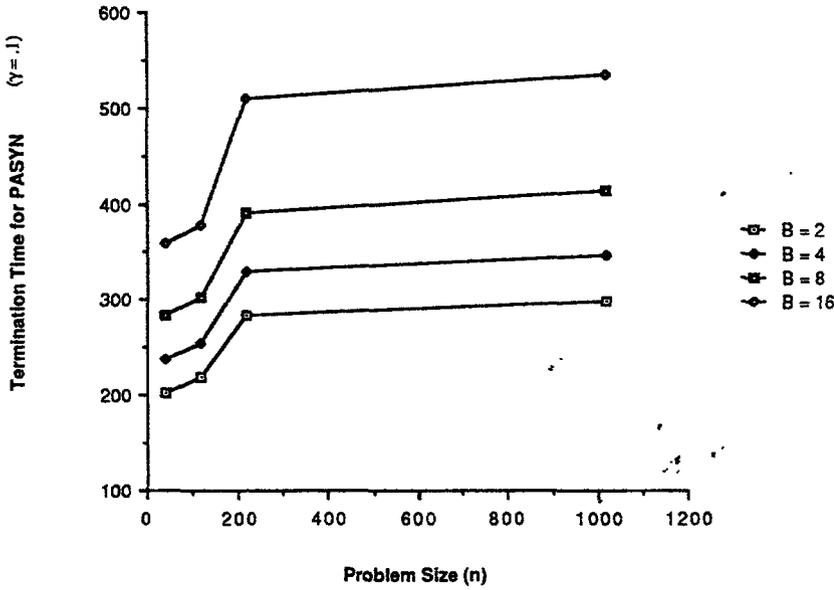


FIG. 8.1(a). Termination time for PASYN ( $\gamma = 0.1$ ), for different values of  $B$  and  $n$ .

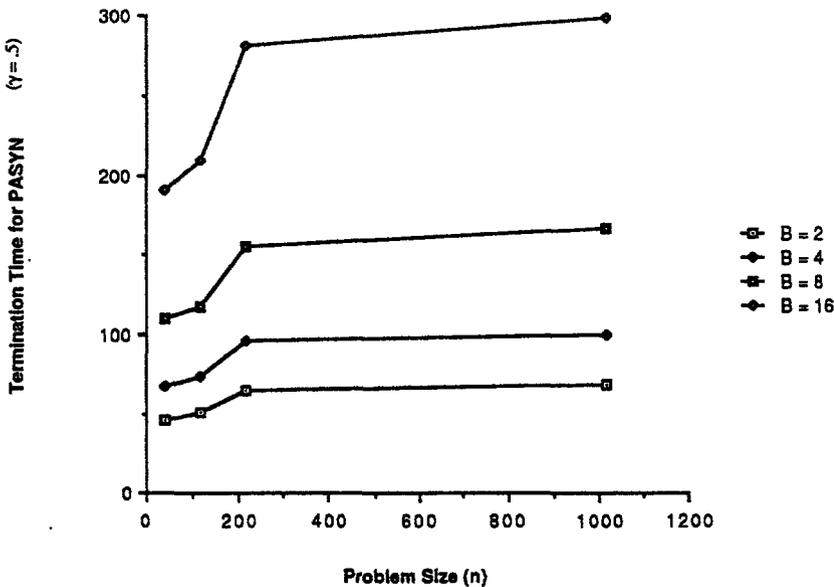


FIG. 8.1(b). Termination time for PASYN ( $\gamma = 0.5$ ), for different values of  $B$  and  $n$ .

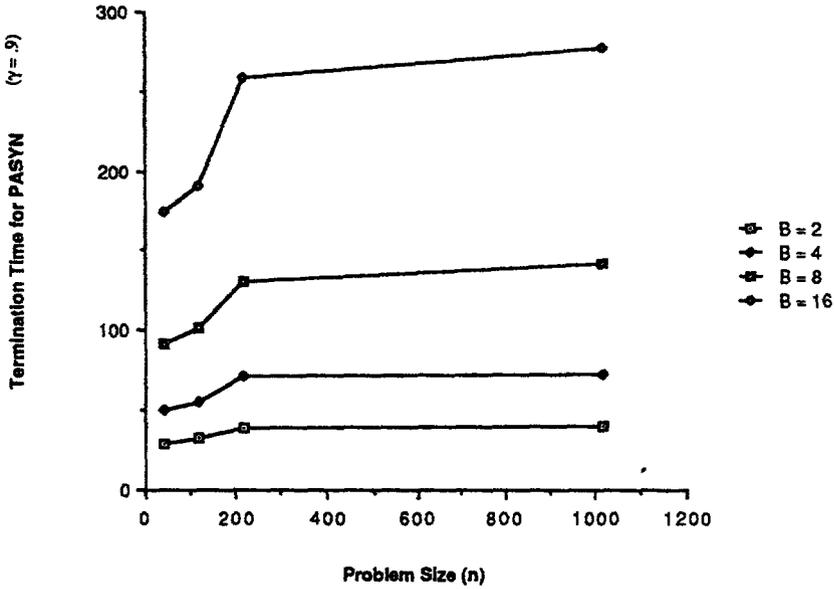


FIG. 8.1(c). Termination time for PASYN ( $\gamma = 0.9$ ), for different values of  $B$  and  $n$ .

The termination time of PASYN, for different values of  $\gamma$ ,  $B$ , and  $n$ , is shown in Figs. 8.1(a)–(c). In general, the rate of convergence of PASYN is the fastest for  $\gamma$  near 1 and for  $B$  small, corroborating our intuition. The termination time grows quite slowly with the size of the problem  $n$  but quite fast with decreasing  $\gamma$ . For  $\gamma$  near 1, the termination time grows roughly linearly with  $B$  (but not when  $\gamma$  is near 0).

**8.3. An alternative partially asynchronous algorithm.** Consider the function  $f^0: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  whose  $i$ th component is given by

$$(8.3) \quad f_i^0(x) = \begin{cases} h_i(x) & \text{if } i \neq 1, \\ x_1 & \text{otherwise.} \end{cases}$$

It is shown in [5] that the algorithm  $x := f^0(x)$  converges under the total asynchronism assumption. Hence it is of interest to compare this algorithm with that described in § 8.2 (namely PASYN) under the same assumption of partial asynchronism. The partially asynchronous version of the algorithm  $x := f^0(x)$ , called TASYN, is identical to PASYN except that the function  $f$  in (8.2) is replaced by  $f^0$ . (Note that TASYN has the advantage that it uses a unity stepsize.)

The termination time of TASYN, for different values of  $B$  and  $n$ , is shown in Fig. 8.2. A comparison with Figs. 8.1(a)–(c) shows that TASYN is considerably slower than PASYN. The speed of TASYN is improved if  $f$  in (8.2) is replaced by  $f^0$  only after a certain amount of time has elapsed, but the improvement is still not sufficient for it to compete with PASYN.

**8.4. Two synchronous algorithms.** In this subsection we consider two types of synchronous algorithms based on  $h$ : the Jacobi algorithm and the Gauss-Seidel algorithm. In particular, the Gauss-Seidel algorithm has been shown to be efficient for practical computation (see [6], [9], [21], [31]). Hence, by comparing the asynchronous algorithms with these algorithms, we can better measure the practical efficiency of the former.

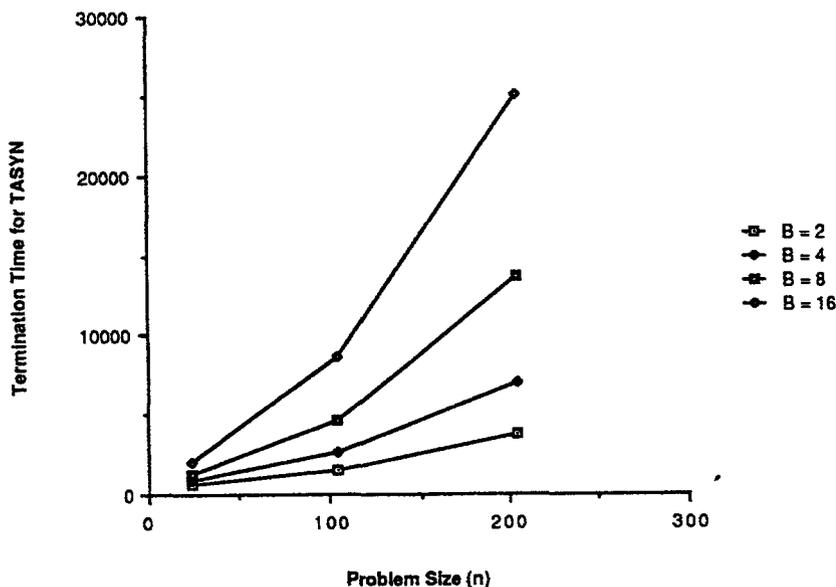


FIG. 8.2. Termination time for TASYN, for different values of  $B$  and  $n$ .

The Jacobi algorithm, called SYNJB, is a parallel algorithm that generates a sequence  $\{x(t)\}$  according to

$$x(t+1) = (1 - \gamma)x(t) + \gamma h(x(t)),$$

where  $\gamma \in (0, 1)$ . The initial estimates  $x_1(0), \dots, x_n(0)$  are independently generated from a uniform distribution over the interval  $[0, 10]$ , and the algorithm terminates at time  $t$  if  $\|x(t) - x(t-1)\| \leq 0.001$ . (SYNJB can be seen to be a special case of PASYNB where  $B = 1$  and hence  $\{x(t)\}$  converges to a fixed point of  $h$ .)

Consider any positive integer  $b$  and any function  $\beta: \{1, \dots, n\} \rightarrow \{1, \dots, b\}$  such that  $h_i(x)$  does not depend on  $x_j$  if  $\beta(i) = \beta(j)$ . We associate with  $b$  and  $\beta$  a Gauss-Seidel algorithm that generates a sequence  $\{x(t)\}$  according to

$$x_i(t+1) = \begin{cases} h_i(x(t)) & \text{if } t \equiv \beta(i) - 1 \pmod{b}, \\ x_i(t) & \text{otherwise.} \end{cases}$$

In our simulation, the initial estimates  $x_1(0), \dots, x_n(0)$  are independently generated from a uniform distribution over the interval  $[0, 10]$  and the algorithm terminates at time  $t$  if

$$\max_{\tau, \tau' \in \{t-b, \dots, t\}} \|x(\tau) - x(\tau')\| \leq 0.001.$$

(Convergence of  $\{x(t)\}$  to a fixed point of  $h$  follows from Proposition 2.4 in [6]. Note that, similar to TASYN, this algorithm has the advantage of using a unity stepsize.) We consider both a serial and a parallel version of this algorithm (this is done by choosing  $b$  and  $\beta$  appropriately). SYNGS1 is the serial version which chooses  $b = n$  and  $\beta(i) = i$  for all  $i$ . SYNGS2 is the parallel version which uses a coloring heuristic to find, for each problem, a choice of  $b$  and  $\beta$  for which  $b$  is small.

The termination time for SYNJB, SYNGS1 and SYNGS2, for different values of  $n$ , are shown in Figs. 8.3(a)-(b). In Fig. 8.3(a), the choice of  $b$  obtained by the coloring heuristic in SYNGS2 is also shown (in parentheses). In general, SYNJB is considerably

faster than either of the two Gauss-Seidel algorithms SYNGS1 and SYNG2 (however in SYNJB all processors must compute at all times). From Fig. 8.3(b) we see that, as  $n$  increases and the problems become more sparse, SYNGS2 (owing to its high parallelism) becomes much faster than the serial algorithm SYNGS1. (Notice that the time for SYNGS1 is approximated by the time for SYNGS2 multiplied by  $n/b$ , as expected.) Comparing Fig. 8.3(a) with Fig. 8.1(c), we see that SYNJB is approximately

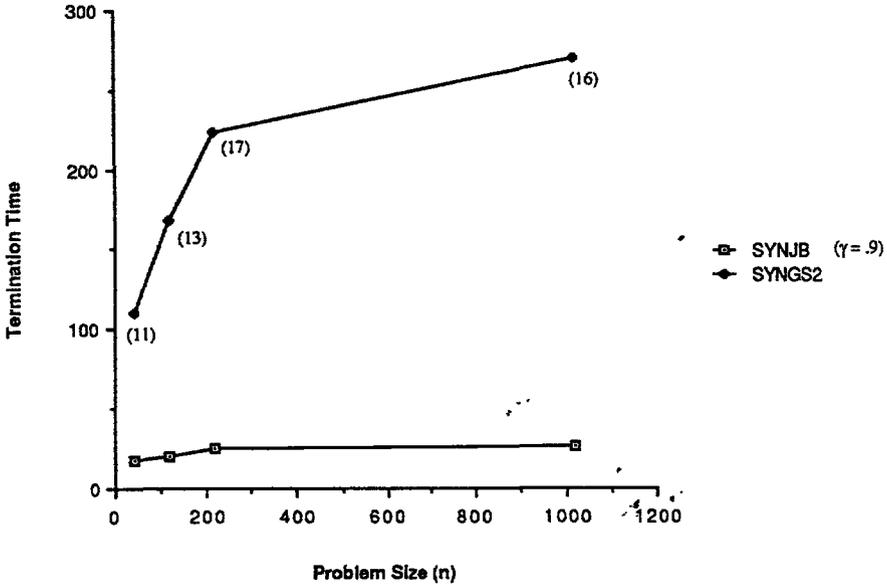


FIG. 8.3(a). Comparing the termination time for the two synchronous, parallel algorithms SYNJB ( $\gamma = 0.9$ ) and SYNGS2, for different values of  $n$ .

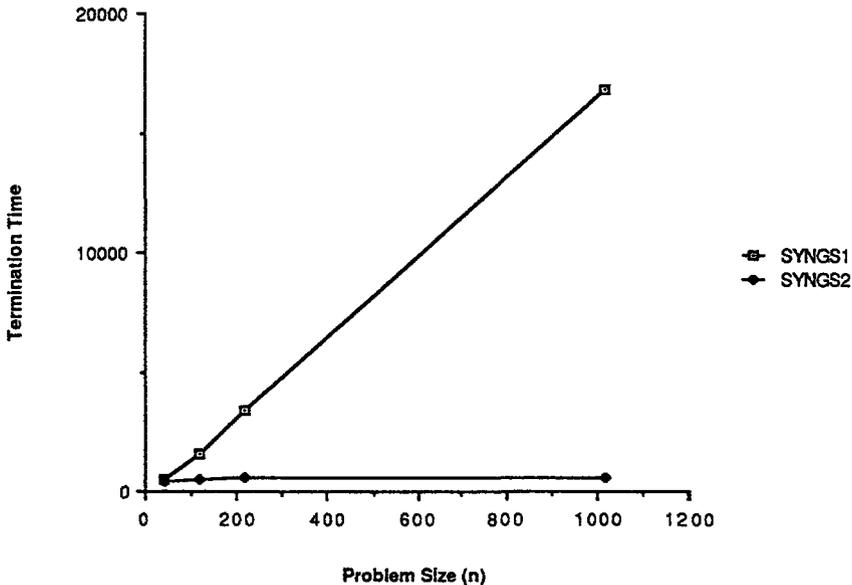


FIG. 8.3(b). Comparing the termination time for the serial algorithm SYNGS1 and for the synchronous, parallel algorithm SYNGS2, for different values of  $n$ .

3/2 times faster than PASYN and that PASYN is faster than SYNGS2, unless PASYN suffers long delays.

**8.5. Simulation of synchronous algorithms in the face of communication delays.** In this subsection we consider the execution of the synchronous iterations of § 8.4 in an asynchronous computing environment, that is, in an environment where communication delays are variable and unpredictable. The mathematical description of the algorithms in this subsection is identical to that of the algorithms considered in the preceding subsection; for this reason, the number of iterations until termination is also the same. On the other hand, each processor must wait until it receives the updates of the other processors before it can proceed to the next iteration. For this reason, the actual time until termination is different from the number of iterations. In our simulation, the delays are randomly generated but their statistics are the same as in our simulation of asynchronous algorithms in §§ 8.2 and 8.3 (uniformly distributed over the set  $\{0, 1, \dots, B-1\}$ , where  $B$  denotes the maximum delay). This will allow us to determine whether asynchronous methods are preferable in the face of communication delays.

More precisely, consider any synchronous algorithm and let  $T$  denote the number of iterations at which this algorithm terminates. With each  $t \in \{1, \dots, T\}$  and each  $i \in \{1, \dots, n\}$ , we associate a positive integer  $\sigma_i(t)$  to represent the "time" at which the update of the  $i$ th component at iteration  $t$  is performed in the corresponding asynchronous execution. (Here we distinguish between "iteration" for the synchronous algorithm and "time" for the asynchronous execution.) Then  $\{\sigma_i(t)\}$  is recursively defined by the following formula:

$$\sigma_i(t) = \max \{ \sigma_j(t-1) + (\text{communication delay from proc. } j \text{ to proc. } i \text{ at time } \sigma_j(t-1)) \},$$

where the maximization is taken over all  $j$  such that the  $j$ th component influences the  $i$ th component at iteration  $t$ . The termination time of the asynchronous algorithm is then taken to be

$$\max_i \{ \sigma_i(T) \}.$$

The partially asynchronous algorithms that simulate SYNJB, SYNGS1 and SYNGS2 are called, respectively, PASYNJB, PASYNGS1 and PASYNGS2. The termination times for these algorithms are shown in Figs. 8.4–8.6 (they are obtained from the termination times shown in Figs. 8.3(a)–(b) using the procedure described above). Comparing these figures with Figs. 8.1(a)–(c), we see that PASYNJB is roughly 3/4 as fast as PASYN (when both use the same stepsize  $\gamma = 0.9$ ) while the other two algorithms PASYNGS1 and PASYNGS2 are considerably slower than PASYN (even when PASYN uses the most conservative stepsize  $\gamma = 0.1$ ).

To summarize, we can conclude that PASYN is the fastest algorithm for partially asynchronous computation and that its synchronous counterpart SYNJB is the fastest for synchronous parallel computation. We remark that similar behavior was observed in other network flow problems that were generated. Furthermore, the asynchronous algorithm PASYN seems to be preferable to its synchronous counterpart SYNJB in the face of delays. In practice, the assumption that the delays are independent and identically distributed might be violated. For example, queueing delays are usually dependent; also, the distance between a pair of processors who need to communicate could be variable, in which case the delays are not identically distributed. On the other hand, such aspects cannot be simulated convincingly without having a particular parallel computing system in mind.

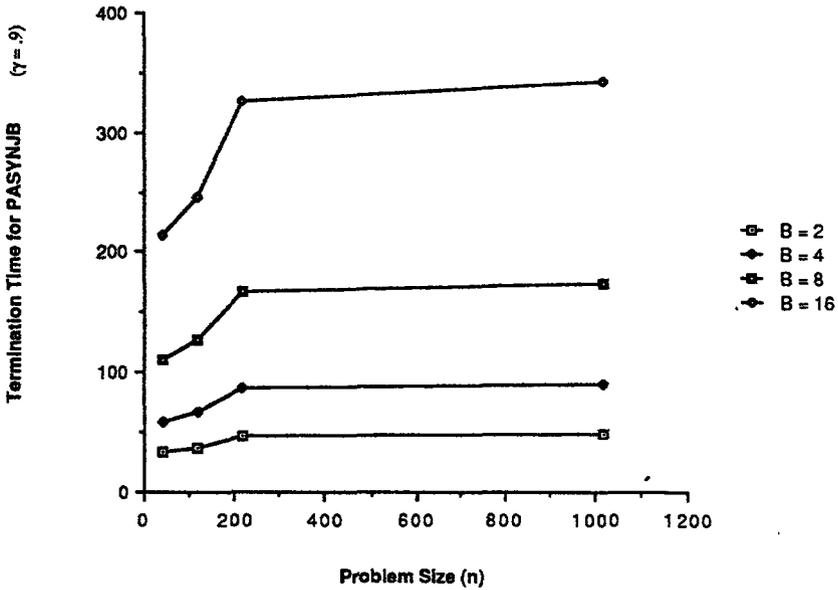


FIG. 8.4. Termination time for PASYNJB ( $\gamma = 0.9$ ), for different values of  $B$  and  $n$ .

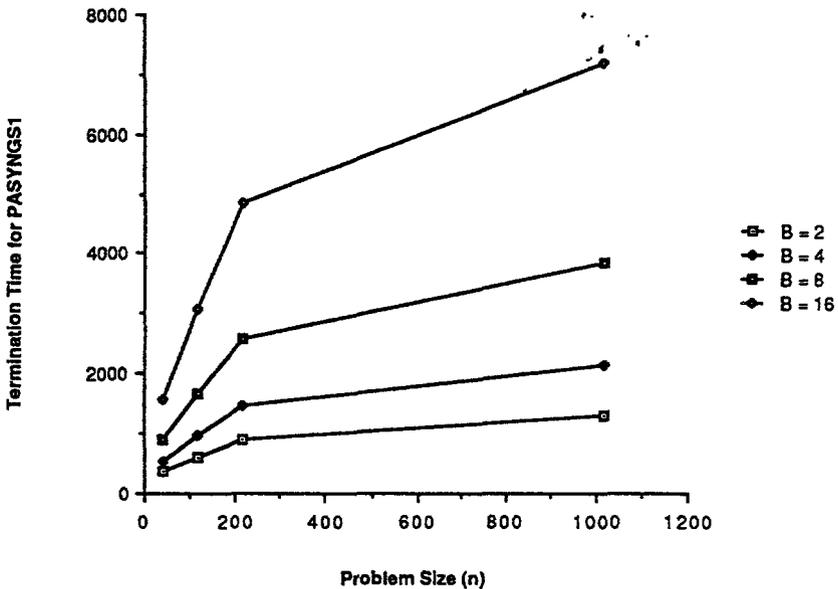


FIG. 8.5. Termination time for PASYNGB1, for different values of  $B$  and  $n$ .

**9. Conclusion and extensions.** In this paper we have presented a general framework, based on nonexpansive mappings, for partially asynchronous computation. The key to this framework is a new class of functions that are nonexpansive with respect to the maximum norm. We showed that any algorithm whose algorithmic mapping belongs to this class converges under the partial asynchronism assumption with an arbitrarily large bound on the delays. While some of the asynchronous algorithms thus obtained are known, others are quite new. Numerical experimentation

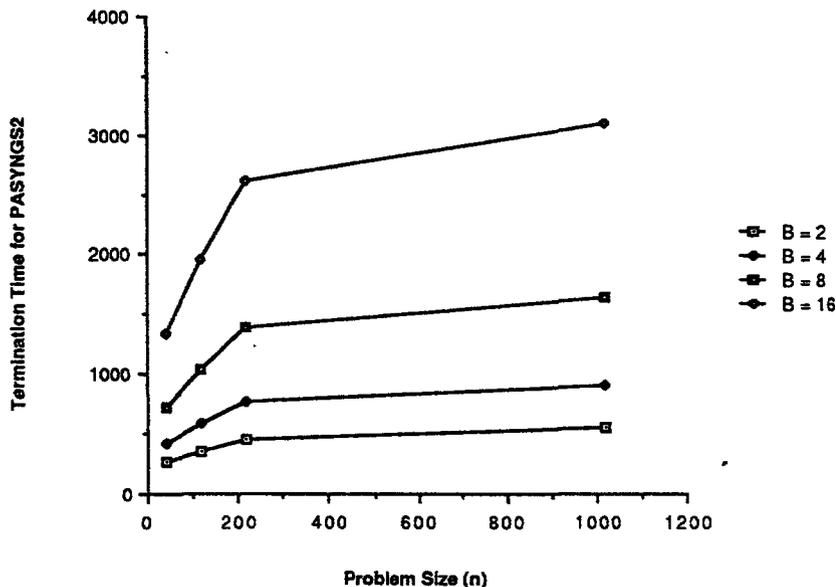


FIG. 8.6. Termination time for PASYNGS2, for different values of  $B$  and  $n$ .

with network flow problems suggests that, for partially asynchronous computation, the new algorithms may be substantially faster than those obtained from synchronous algorithms.

#### REFERENCES

- [1] G. M. BAUDET, *Asynchronous iterative methods for multiprocessors*, *J. Assoc. Comput. Mach.*, 15 (1978), pp. 226-244.
- [2] D. P. BERTSEKAS, *Distributed dynamic programming*, *IEEE Trans. Automat. Control*, AC-27 (1982), pp. 610-616.
- [3] ———, *Distributed asynchronous computation of fixed points*, *Math. Programming*, 27 (1983), pp. 107-120.
- [4] ———, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [5] D. P. BERTSEKAS AND D. EL BAZ, *Distributed asynchronous relaxation methods for convex network flow problems*, *SIAM J. Control Optim.*, 25 (1987), pp. 74-85.
- [6] D. P. BERTSEKAS, P. HOSEIN, AND P. TSENG, *Relaxation methods for network flow problems with convex arc costs*, *SIAM J. Control Optim.*, 25 (1987), pp. 1219-1243.
- [7] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [8] D. CHAZAN AND W. MIRANKER, *Chaotic relaxation*, *Linear Algebra Appl.*, 2 (1969), pp. 199-222.
- [9] R. W. COTTLE, S. G. DUVALL, AND K. ZIKAN, *A Lagrangean relaxation algorithm for the constrained matrix problem*, *Naval Res. Logist. Quart.*, 33 (1986), pp. 55-76.
- [10] R. W. COTTLE AND A. F. VEINOTT, JR., *Polyhedral sets having a least element*, *Math. Programming*, 3 (1972), pp. 238-249.
- [11] K. DEIMLING, *Nonlinear Functional Analysis*, Springer-Verlag, Berlin, New York, 1985.
- [12] R. S. DEMBO AND J. G. KLINCEWICZ, *A scaled reduced gradient algorithm for network flow problems with convex separable costs*, *Math. Programming Stud.*, 15 (1981), pp. 125-147.
- [13] H. GABBAY, *A note on polyhedral sets having a least element*, *Math. Programming*, 11 (1976), pp. 94-96.
- [14] F. R. GANTMACHER, *The Theory of Matrices*, Vol. II, Chelsea, New York, 1960.
- [15] J. J. HOPFIELD, *Neurons with graded response have collective computational properties like those of two-state neurons*, *Proc. Nat. Acad. Sci. U.S.A.*, 81 (1984), pp. 3088-3092.

- [16] J. J. HOPFIELD AND D. W. TANK, *Computing with neural circuits: a model*, *Science*, 233 (1986), pp. 625-633.
- [17] K. R. JAMES, *Convergence of matrix iterations subject to diagonal dominance*, *SIAM J. Numer. Anal.*, 10 (1973), pp. 478-484.
- [18] D. KLINGMAN, A. NAPIER, AND J. STUTZ, NETGEN—*A program for generating large scale (un)capacitated assignment, transportation and minimum cost flow network problems*, *Management Sci.*, 20 (1974), pp. 814-822.
- [19] R. P. LIPPMANN, *An introduction to computing with neural nets*, *IEEE ASSP Magazine*, (1987), pp. 4-22.
- [20] B. LUBACHEVSKY AND D. MITRA, *A chaotic, asynchronous algorithm for computing the fixed point of a nonnegative matrix of unit spectral radius*, *J. Assoc. Comput. Mach.*, 33 (1986), pp. 130-150.
- [21] A. OHUCHI AND I. KAJI, *Lagrangian dual coordinatewise maximization algorithm for network transportation problems with quadratic costs*, *Networks*, 14 (1984), pp. 515-530.
- [22] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
- [23] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
- [24] ———, *Network Flows and Monotropic Optimization*, Wiley-Interscience, New York, 1984.
- [25] T. J. SEJNOWSKI, *Open questions about computation in cerebral cortex*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. II, J. L. McClelland, D. E. Rumelhart, and the PDP Research Group, eds., MIT Press, Cambridge, MA, 1986, pp. 372-389.
- [26] P. TSENG, *Distributed computation for linear programming problems satisfying a certain diagonal dominance condition*, Working Paper 1256, Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, Canada; also LIDS Report, MIT, Cambridge, MA, December 1986; *Math. Oper. Res.*, to appear.
- [27] J. N. TSITSIKLIS AND D. P. BERTSEKAS, *Distributed asynchronous optimal routing in data networks*, *IEEE Trans. Automat. Control*, AC-31 (1986), pp. 325-332.
- [28] J. N. TSITSIKLIS, D. P. BERTSEKAS, AND M. ATHANS, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, *IEEE Trans. Automat. Control*, AC-31 (1986), pp. 803-812.
- [29] J. N. TSITSIKLIS, *Problems in decentralized decision making and computation*, Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1984.
- [30] ———, *On the stability of asynchronous iterative processes*, *Mathematical Systems Theory*, 20 (1987), pp. 137-153.
- [31] S. A. ZENIOS AND J. M. MULVEY, *Relaxation techniques for strictly convex network problems*, *Ann. Oper. Res. 5: Algorithms and Software for Optimization* C. L. Monma, ed., Baltzer, Basel, Switzerland, 1986, pp. 517-538.
- [32] S. A. ZENIOS AND J. M. MULVEY, *A distributed algorithm for convex network optimization problems*, *Parallel Comput.*, 6 (1988), pp. 45-56.
- [33] S. A. ZENIOS AND R. A. LASKEN, *Nonlinear network optimization on a massively parallel connection machine*, in *Ann. Oper. Res. 14: Parallel Optimization on Novel Computer Architectures*, R. R. Meyer and S. A. Zenios, eds., Baltzer, Basel, Switzerland (1988), pp. 147-165.