Parallel Gradient Distribution in Unconstrained Optimization

O. L. Mangasarian*

SIAM Journal on Optimization 33, 1995, 1916-1925

Abstract

A parallel version is proposed for a fundamental theorem of serial unconstrained optimization. The parallel theorem allows each of k parallel processors to use simultaneously a different algorithm, such as a descent, Newton, quasi-Newton or a conjugate gradient algorithm. Each processor can perform one or many steps of a serial algorithm on a portion of the gradient of the objective function assigned to it, independently of the other processors. Eventually a synchronization step is performed which, for differentiable convex functions, consists of taking a strong convex combination of the k points found by the k processors. A more general synchronization step, applicable to convex as well as nonconvex functions, consists of taking the best point found by the k processors or any point that is better. The fundamental result that we establish is that any accumulation point of the parallel algorithm is stationary for the nonconvex case, and is a global solution for the convex case. Computational testing on the Thinking Machines CM-5 multiprocessor indicate a speedup of the order of the number of processors employed.

1 Introduction

In this work we are interested in parallel algorithms for solving the unconstrained minimization problem

$$\min_{x\in R^n} f(x) \tag{1}$$

where f is a differentiable function from the n-dimensional real space R^n into R. The basic idea behind our approach is to assign a portion of the gradient ∇f of f, to one of k processors, let each processor perform one or more steps of a serial algorithm on its portion of the gradient, and then synchronize the processors eventually. The synchronization consists of taking a strong convex combination of the k points found by the k processors when f is convex. For nonconvex f, the best point found by the k processors can be taken, or any other point with a lower value of f will work.

The fundamental theorem we intend to parallelize is related to some classical forcing function theorems given in [7, 4, 11] that establish convergence for a wide class of algorithms. Such algorithms typically consist of a direction choice followed by a stepsize choice. The combined direction-stepsize choice generates a decrease in the objective function that forces the eventual satisfaction of an optimality condition, namely the vanishing of the gradient. Direction choices include descent directions, Newton, quasi-Newton and conjugate directions. Stepsize choices along the chosen direction include minimization, finding the first stationary point, interval stepsize, the Armijo

*Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, email: *olvi@cs.wisc.edu.* This material is based on research supported by Air Force Office of Scientific Research Grants AFOSR-89-0410 and F49620-94-1-0036, and National Science Foundation Grants CCR-9101801 and CDA-9024618.

stepsize and others. Related algorithms, wherein the objective function is sequentially minimized with respect to certain variables, include the serial algorithm proposed by Warga [16] for a strictly convex function in each block of variables and in which the function is sequentially minimized for each block of variables, and the coordinate descent methods of Tseng [15] and Luo and Tseng [8]. Other parallelization schemes are discussed extensively in [2].

We note that our parallelization proofs are direct extensions of those for general serial algorithms. However the resulting parallel algorithms are quite general and have significant theoretical and computational implications. For example, the parallelization proposed here played an important role in establishing the convergence and computational results of the parallel backpropagation algorithm of neural networks [9], the parallel variable distribution algorithm for unconstrained and constrained optimization [6], and the parallel multicategory discrimination problem [1].

We give now an outline of the paper. In Section 2 we establish two serial convergent algorithm Theorems 2.1 and 2.2 (SCAT1 and SCAT2) which cover many unconstrained direction-stepsize algorithms that are suitable for parallelization. We also give a number of specific instances of well known algorithms satisfying conditions of these theorems. In Section 3 we establish a number of parallel convergent algorithm theorems that utilize the serial algorithms. In Theorem 3.1 (Convex PCAT1), which covers the convex case, each processor takes one step of any serial algorithm covered by SCAT1 or SCAT2, and then a strong convex combination (positively weighted average) of all the points is taken as the next iterate. Corollary 3.1 (Nonconvex PCAT1) differs from the Convex PCAT1 in that the synchronization step consists of taking the best point found by the k processors, or searching for a better point. By better we mean, of course, lower f value. Corollary 3.2 (Partially Asynchronous Nonconvex PCAT1) allows partial asynchronization among the k processors in the sense that each processor is free to perform any number of steps of the serial algorithm that is desirable (say, until further improvement in each processor is very small), followed by a synchronization step that consists of taking the best point or searching for a better one. Theorem 3.2 (Partially Asynchronous Nonconvex PCAT2) combines, in a similar manner to SCAT2 for the serial case, the direction and stepsize choices of Corollary 3.2 into a simpler and more general forcing function condition (20). However, this theorem is not as suggestive of an explicit computational scheme as the Partially Asynchronous Nonconvex PCAT1 of Corollary 3.2. In the concluding Section 4 we report briefly on computational experience with parallel gradient distribution algorithms on multicategory discrimination problems [1], and on publicly available test problems [6] from the constrained and unconstrained testing environment CUTE [3]. Computations were carried out on the Thinking Machines CM-5 multiprocessor. Speedup efficiency depended on problem size and number of processors employed (2 to 32) and averaged between 129% and 20%.

We now briefly describe our notation. The sequence $\{x_i\},\ i = 0, 1, \ldots,$ will represent iterates in the n-dimensional real space R^n generated by some algorithm. For $\ell = 1, \ldots, k$, $x_i^\ell \in R^{n^\ell}$ will represent an n^ℓ-dimensional subset of components of x_i, where $\sum_{\ell=1}^{k} n^\ell = n$. The complement of ℓ in $\{1, \ldots, k\}$ will be denoted by $\bar\ell$ and we write $x_i = (x_i^\ell,\ x_i^{\bar\ell}),\ \ell = 1, \ldots, k$. For a differentiable function $f\colon R^n \to R$, ∇f will denote the n-dimensional vector of partial derivatives with respect to x, and $\nabla_\ell f$ will denote the n^ℓ-dimensional vector of partial derivatives with respect to $x^\ell \in R^{n^\ell},\ \ell = 1, \ldots, k$. For k points y in R^n, $\sum_{j=1}^{k} \lambda_j y_j$, such that $\lambda_j \ge \delta > 0$ and $\sum_{j=1}^{k} \lambda_j = 1$, is said to be a strong convex combination of the points $y_j,\ j = 1, \ldots, k$. If f has continuous first partial derivatives on R^n, we say $f \in C^1(R^n)$. If f has Lipschitz continuous first partial derivatives on R^n with constant $K > 0$, that is:

$$\|\nabla f(y) - \nabla f(x)\| \le K\|y - x\| \qquad \forall x,\ y \in R^n$$

we write $f \in LC^1_K(R^n)$. Here and throughout, $\|\cdot\|$ denotes the two norm, that is $\|z\| = (z^T z)^{\frac{1}{2}}$, for z in a finite-dimensional real space of unspecified dimension.

2 Serial Convergent Algorithm Theorems

We begin first with a simple serial convergent algorithm theorem (SCAT1) for the solution of the unconstrained minimization problem (1). The theorem is related to some classical forcing function theorems given in [7, 4] that establish convergence for a wide class of algorithms that consist of a direction choice followed by a stepsize choice. The decrease in the objective function forces the satisfaction of an optimality condition, namely the vanishing of the gradient. Before stating and proving SCAT1 we adapt the definition of a forcing function [10, p.479] for our purposes.

Definition 2.1 Forcing function *A continuous function σ from the nonnegative real line R_+ into itself such that $\sigma(0) = 0,\ \sigma(\zeta) > 0$ for $\zeta > 0$ and such that for the sequence of nonnegative real numbers $\{\zeta_i\}$:*

$$\{\sigma(\zeta_i)\} \to 0 \ \text{ implies } \ \{\zeta_i\} \to 0.$$

is said to be a forcing function on the sequence $\{\zeta_i\}$.

Some simple typical examples of forcing functions are

$$\alpha\zeta,\ \alpha\zeta^2,\ \max\{\sigma_1(\zeta),\ \sigma_2(\zeta)\},\ \min\{\sigma_1(\zeta),\ \sigma_2(\zeta)\} \text{ and } \sigma_2(\sigma_1(\zeta)),$$

where α is a positive number and $\sigma_1(\zeta)$ and $\sigma_2(\zeta)$ are forcing functions. We now state and prove SCAT1.

Theorem 2.1 Serial convergent algorithm Theorem 1 (SCAT1) *Let $f \in C^1(R^n)$. Start with any $x_0 \in R^n$. Given x_i, stop if $\nabla f(x_i) = 0$, else compute x_{i+1} from a direction d_i and stepsize λ_i that satisfy:*

Direction d_i:

$$-\nabla f(x_i)^T d_i \ge \sigma_1(\|\nabla f(x_i)\|), \tag{2}$$

where σ_1 is a forcing function on $\{\|\nabla f(x_i)\|\}$, and

Stepsize λ_i:

$$x_{i+1} = x_i + \lambda_i d_i \tag{3}$$

such that

$$f(x_i) - f(x_{i+1}) \ge \sigma_2(-\nabla f(x_i)^T d_i) \ge 0, \tag{4}$$

where σ_2 is a forcing function on the sequence of nonnegative real numbers $\{-\nabla f(x_i)^T d_i\}$ for bounded $\{d_i\}$. Then either $\{x_i\}$ terminates at a stationary point $x_{\bar{i}}$, that is $\nabla f(x_{\bar{i}}) = 0$, or $\nabla f(\bar{x}) = 0$ for each accumulation point $(\bar{x}, \bar{d})$ of the sequence $\{x_i, d_i\}$.

Proof The algorithm terminates at an $x_{\bar{i}}$ only if $\nabla f(x_{\bar{i}}) = 0$. Suppose now it does not terminate and that $\{(x_{i_j},\ d_{i_j})\} \to (\bar{x},\ \bar{d})$. Since f is continuous, $\lim_{j\to\infty} f(x_{i_j}) = f(\bar{x})$. By the stepsize condition

(4), the sequence $\{f(x_i)\}$ is nonincreasing and has an accumulation point $f(\bar{x})$, and hence converges to $f(\bar{x})$. By (4) and the continuity of $\sigma_2(\zeta)$

$$0 = \lim_{j\to\infty} f(x_{i_j}) - f(x_{i_j+1}) \geq \lim_{j\to\infty} \sigma_2(-\nabla f(x_{i_j})^T d_{i_j}) = \sigma_2(-\nabla f(\bar{x})^T \bar{d}) \geq 0.$$

Hence $\nabla f(\bar{x})^T \bar{d} = 0$. But by the direction condition (2)

$$0 = -\nabla f(\bar{x})^T \bar{d} = -\lim_{j\to\infty} \nabla f(x_{i_j})^T d_{i_j} \geq \lim_{j\to\infty} \sigma_1(\|\nabla f(x_{i_j})\|) = \sigma_1(\|\nabla f(\bar{x})\|) \geq 0.$$

Hence $\nabla f(\bar{x}) = 0$. $\square$

We note that the boundedness condition on $\{d_i\}$, which does not restrict Theorem 2.1, was not explicitly used in the proof. However, this condition simplifies the application of the theorem to specific stepsize choices, such as the first stationary-point and Armijo stepsize choices given below.

We give now examples of direction and stepsize choices that satisfy the assumptions of Theorem 2.1.

Example 2.1 Serial direction choices *For $f \in C^1(R^n)$ and σ a forcing function, a direction $d_i \in R^n$ satisfying any of the following conditions will satisfy condition (2):*

(i) **Descent direction**

$$-d_i^T \nabla f(x_i) \geq \alpha \|\nabla f(x_i)\|^\beta \text{ for some } \alpha > 0,\ \beta > 0.$$

(ii) **Quasi-Newton direction**

$$d_i = -H_i \nabla f(x_i),\ H_i \in R^{n\times n},\ z^T H_i z \geq \alpha \|z\|^2 \qquad \forall z \in R^n, \text{ for some } \alpha > 0.$$

(iii) **Conjugate direction**

$$d_i = -\nabla f(x_i) + \alpha_i d_{i-1}$$

$$\frac{\|\nabla f(x_i)\|^2}{\|\nabla f(x_i)\| + |\alpha_i|\|d_{i-1}\|} \geq \sigma(\|\nabla f(x_i)\|), \tag{5}$$

where σ is a forcing function on $\{\|\nabla f(x_i)\|\}$.

We note that the conjugate direction conditions of (iii) are satisfied by the Polyak-Polak-Ribière [13, 12, 14, 11] coefficient

$$\alpha_i := \frac{(\nabla f(x_i) - \nabla f(x_{i-1}))^T \nabla f(x_i)}{\|\nabla f(x_{i-1})\|^2} \tag{6}$$

for $f \in C^2(R^n)$ and such that

$$\beta ||z||^2 \geq z^T \nabla^2 f(x) z \geq \alpha \|z\|^2 \qquad \forall z \in R^n \text{ for some } \beta \geq \alpha > 0. \tag{7}$$

We also note that the Newton direction $d_i = -\nabla^2 f(x_i)^{-1} \nabla f(x_i)$ satisfies (ii) above under the same condition (7).

We give now stepsize choices that satisfy conditions (3)-(4) of Theorem 2.1.

Example 2.2 Serial stepsize choices. *For $d_i \in R^n$ and $f \in C^1(R^n)$, a $\lambda_i \geq$ satisfying any one of the following conditions will satisfy conditions (3)-(4) of Theorem 2.1:*

(i) **Minimum along d_i**

$$\lambda_i \in \arg\min_{\lambda \geq} f(x_i + \lambda d_i),\ f \in LC^1_K(R^n).$$

(ii) **First stationary point**

$$\lambda_i \in \arg\min_{\lambda \geq} \{\lambda | \nabla f(x_i + \lambda d_i)^T d_i = 0\},\ f \in LC^1_K(R^n).$$

(iii) **Interval stepsize**

$$0 < \varepsilon_1 \leq \lambda_i \leq \frac{2}{\rho K} - \varepsilon_2,\ \|d_i\|^2 \leq -\rho \nabla f(x_i)^T d_i,\ f \in LC^1_K(R^n) \text{ for some } \varepsilon_1 > 0,\ \varepsilon_2 > 0 \text{ and } \rho > 0.$$

(iv) **Armijo** *[5, pp.118-119]*

$\lambda_i = \max\{\bar{\lambda}_i, \frac{\bar{\lambda}_i}{2}, \ldots\}$ *such that*

$$f(x_i) - f(x_i + \lambda_i d_i) \geq -\delta \lambda_i \nabla f(x_i)^T d_i \quad \textit{for some } \delta \in (0,1),$$

and $\bar{\lambda}_i \geq \dfrac{\sigma(-\nabla f(x_i)^T d_i)}{-\nabla f(x_i)^T d_i}$, *where σ is a forcing function and $f \in LC^1_K(R^n)$.*

It takes a bit of algebra to show that each of the four stepsizes (i) to (iv) above satisfy conditions (3)-(4) of Theorem 2.1. We omit the details here.

We note that Theorem 2.1 can be written in a more general and simpler, but algorithmically less suggestive, form by combining conditions (2) and (4) into the single condition (8) below. This results in the following theorem, the proof of which either follows from that of Theorem 2.1 or can be given in a few lines as is done below.

Theorem 2.2 Serial convergent algorithm theorem 2 (SCAT2) *Let $f \in C^1(R^n)$. Start with any $x_0 \in R^n$. Given x_i, stop if $\nabla f(x_i) = 0$, else determine x_{i+1} such that*

$$f(x_i) - f(x_{i+1}) \geq \sigma(\|\nabla f(x_i\|), \tag{8}$$

where σ is a forcing function on $\{\|\nabla f(x_i)\|\}$. Then either $\{x_i\}$ terminates at a stationary point $x_{\bar{i}}$, or each accumulation point $\bar{x}$ of $\{x_i\}$ is stationary.

Proof Suppose $\nabla f(x_i) \neq 0$ for all i and that $\{x_{i_j}\}$ converges to $\bar{x}$. Since the nonincreasing sequence $\{f(x_i)\}$ has an accumulation $f(\bar{x})$, it converges to $f(\bar{x})$. By (8) we have that

$$0 = \lim_{j\to\infty} (f(x_{i_j}) - f(x_{i_j+1})) \geq \lim_{j\to\infty} \sigma(\|\nabla f(x_{i_j})\|) \geq 0.$$

Hence $\lim_{j\to\infty} \|\nabla f(x_{i_j})\| = 0$ and $\nabla f(\bar{x}) = 0$. □

We note that the full sequences $\{f(x_i)\}$ and $\{\|\nabla f(x_i)\|\}$ converge if f is bounded below. We state this as the following corollary.

Corollary 2.1 Function and gradient convergence *Let f be bounded below on the level set $S(x_0) = \{x | f(x) \leq f(x_0)\}$. Then the sequence $\{f(x_i)\}$ of Theorems 2.1 and 2.2 converges, and the* $\lim_{i\to\infty} \|\nabla f(x_i)\| = 0$.

Proof From (8) the sequence $\{f(x_i)\}$ is nonincreasing, and since $\{x_i\}$ remains in $S(x_0)$, $\{f(x_i)\}$ is bounded below and hence converges. From (8), we have that $\lim_{i\to\infty} \sigma(\|\nabla f(x_i)\|) = 0$, and hence $\lim_{i\to\infty} \|\nabla f(x_i)\| = 0$. □

We now proceed to establish parallel versions of Theorems 2.1 and 2.2 and other parallel results.

3 Parallel Convergent Algorithm Theorems

We shall establish in this section parallel versions of Theorems 2.1 and 2.2. The import of these theorems, PCAT1 and PCAT2, is that they enable each of k processors to perform, on a portion of the gradient that is assigned to it, one or more iterations of the serial algorithms independently of the other processors. The processor picks a direction and stepsize based on the partial gradient assigned to it. A simple synchronization step follows in which a new point is generated by a strong convex combination of the k points obtained by the k processors for the convex case, and by using the best, or better, point obtained by the k processors for the nonconvex case. We first state and prove Theorem 3.1, our parallel theorem for the convex case. Corollary 3.1 extends Theorem 3.1 to the nonconvex case. Corollary 3.2 further extends Corollary 3.1 by allowing partial asynchronization by letting each processor take as many steps as desirable. Finally Theorem 3.2 gives a more general version of Theorem 3.1 for the nonconvex case. We note here that a referee pointed out that the distribution of the gradient can also be made with respect to subspaces induced by other decompositions of R^n. For example, the iterate x_i can be decomposed into $x_i^\ell = P_\ell x_i,\ \ell = 1, \ldots, k$, instead of into subvectors x_i^ℓ of x_i. Here $P_1, \ldots, P_k$ are projection matrices (that is $P_i^2 = P_i,\ P_i^T = P_i,\ i = 1, \ldots, k$) such that $\sum_{i=1}^k P_i = I$.

Theorem 3.1 Convex parallel convergent algorithm theorem 1 (Convex PCAT1) *Let $f \in C^1(R^n)$ be convex on R^n. Start with any $x_0 \in R^n$. Given x_i stop if $\nabla f(x_i) = 0$, else compute x_{i+1} from directions $d_i^\ell \in R^{n^\ell}$, and stepsizes $\lambda_i^\ell \in R,\ \ell = 1, \ldots, k,\ \sum_{\ell=1}^k n^\ell = n$, that satisfy:*

Direction d_i^ℓ:

$$-\nabla_\ell f(x_i)^T d_i^\ell \geq \tau_\ell(\|\nabla_\ell f(x_i)\|),\ \ell = 1, \ldots k, \tag{9}$$

where τ_ℓ is a forcing function on $\{\|\nabla_\ell f(x_i)\|\}$, $\ell = 1, \ldots, k$.

Stepsize λ_i^ℓ:

Choose $\lambda_i^\ell,\ \ell = 1, \ldots, k$ such that for $\bar{\ell}$, the complement of ℓ in $\{1, \ldots, k\}$:

$$f(x_i) - f(x_i^\ell + \lambda_i^\ell d_i^\ell,\ x_i^{\bar{\ell}}) \geq \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell) \geq, \qquad \ell = 1, \ldots, k, \tag{10}$$

where μ_ℓ is a forcing function on the sequence of nonnegative real numbers $\{-\nabla_\ell f(x_i)^T d_i^\ell\}$ for bounded $\{d_i^\ell\}$, $\ell = 1, \ldots, k$, and

Synchronization:

$$x_{i+1}^\ell = x_i^\ell + \nu_i^\ell \lambda_i^\ell d_i^\ell \qquad \ell = 1, \ldots, k \tag{11}$$

$$\sum_{\ell=1}^k \nu_i^\ell = 1,\ \nu_i^\ell \geq \delta > 0,\ \ell = 1, \ldots, k. \tag{12}$$

Then, either $\{x_i\}$ terminates at a solution $x_{\bar{i}}$ of (1), or for each accumulation point $(\bar{x}, \bar{d})$ of $\{x_i, d_i\}$, $\bar{x}$ is a solution of (1).

Proof We first show that the sequence $\{f(x_i)\}$ is nonincreasing.

$$\begin{aligned}
f(x_i) \quad - \quad & f(x_{i+1}) \\
= \quad & f(x_i^1, \ldots, x_i^k) - f(x_i^1 + \nu_i^1\lambda_i^1 d_i^1, \ldots, x_i^k + \nu_i^k\lambda_i^k d_i^k) \\
= \quad & f(x_i^1, \ldots, x_i^k) - f(\nu_i^1(x_i^1 + \lambda_i^1 d_i^1) + (\sum_{\ell=2}^{k} \nu_i^\ell) x_i^1, \ldots, \nu_i^k(x_i^k + \lambda_i^k d_i^k) + (\sum_{\ell=1}^{k-1} \nu_i^\ell) x_i^k) \\
\geq \quad & \nu_i^1[f(x_i^1, \ldots, x_i^k) - f(x_i^1 + \lambda_i^1 d_i^1,\, x_i^2, \ldots\, x_i^k)] + \ldots \\
& \quad \ldots + \nu_i^k[f(x_i^1, \ldots,\, x_i^k) - f(x_i^1, \ldots,\, x_i^{k-1},\, x_i^k + \lambda_i^k d_i^k)] \quad \text{(By convexity of } f\text{)} \\
\geq \quad & \delta\sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell). \qquad \text{(By (12) and (10).)} \qquad (13)
\end{aligned}$$

Hence

$$f(x_i) - f(x_{i+1}) \geq \delta\sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell) \geq 0, \qquad (14)$$

and the sequence $\{f(x_i)\}$ is nonincreasing.

Now the sequence $\{x_i\}$ of the PCAT1 algorithm terminates only if $\nabla f(x_{\bar{i}}) = 0$, in which case $x_{\bar{i}}$ solves (1). Suppose now that it does not terminate and that $\{(x_{i_j}, d_{i_j})\} \to (\bar{x}, \bar{d})$. Since f is continuous, $\lim_{j\to\infty} f(x_{i_j}) = f(\bar{x})$. Hence the nonincreasing sequence $\{f(x_i)\}$ has an accumulation point $f(\bar{x})$, and consequently the sequence converges to $f(\bar{x})$. By (14) and the continuity of μ_ℓ, $\ell = 1, \ldots, k$, we have that

$$0 = \lim_{j\to\infty}(f(x_{i_j}) - f(x_{i_j+1})) \geq \delta \lim_{j\to\infty} \sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(x_{i_j})^T d_{i_j}^\ell) = \delta\sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(\bar{x})^T \bar{d}^\ell) \geq 0.$$

Hence $\nabla f(\bar{x})^T\bar{d} = 0$. But by the direction condition (9)

$$0 = \nabla f(\bar{x})^T\bar{d} = -\lim_{j\to\infty} \nabla f(x_{i_j})^T d_{i_j} \geq \lim_{j\to\infty} \sum_{\ell=1}^{k} \tau_\ell(\|\nabla f_\ell(x_{i_j})\|) = \sum_{\ell=1}^{k} \tau_\ell(\|\nabla f_\ell(\bar{x})\|) \geq .$$

Hence $\nabla_\ell f(\bar{x}) = 0,\ \ell = 1, \ldots, k$ and consequently $\nabla f(\bar{x}) = 0$ and $\bar{x}$ solves (1). □

We note that the convexity of f was needed in (13) in the proof above, as well as to show that the stationary point generated by PCAT1 is a global solution of $\min_{x\in R^n} f(x)$. However, it is easy to extend Theorem 3.1 to nonconvex f by changing the synchronization procedure (11)-(12) to one that takes the best of the points found by the k processors or a better point. We state this as the following corollary.

Corollary 3.1 Nonconvex parallel convergence algorithm theorem 1 (Nonconvex PCAT1) *Theorem 3.1 holds for nonconvex f, with a resulting stationary point, if the synchronization procedure (11)-(12) is replaced by the following:*

Synchronization:

Find x_{i+1} such that

$$f(x_{i+1}) \leq \min_{1\leq\ell\leq k} f(x_i^\ell + \lambda_i^\ell d_i^\ell,\, x_i^{\bar{\ell}}). \qquad (15)$$

Proof The only changes needed in the proof of Theorem 3.1 in order to apply it here are the following. Replace δ by $\frac{1}{k}$ in (14) and replace the string of inequalities of (13), which establish the monotonicity of $\{f(x_i)\}$ through the convexity of f, by the following:

$$\begin{aligned} f(x_i) - f(x_{i+1}) &\geq \frac{1}{k}\left[f(x_i) - f(x_i^1 + \lambda_i^1 d_i^1,\, x_1^2, \ldots, x_i^k)\right] + \ldots \\ &\qquad \ldots + \frac{1}{k}\left[f(x_i) - f(x_i^1, \ldots, x_i^{k-1},\, x_i^k + \lambda_i^k d_i^k)\right] \quad \text{(By (15))} \\ &\geq \frac{1}{k}\sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell). \end{aligned}$$

□

We note now that partial asynchronization of the k processors for the nonconvex PCAT1 is possible by allowing each of the k processors to take as many steps as desired until, say, they encounter slow convergence, provided we terminate each processor ℓ, $\ell = 1, \ldots, k$, at a point $(y_i^\ell, x_i^{\bar{\ell}})$ such that

$$f(y_i^\ell, x_i^{\bar{\ell}}) \leq f(x_i^\ell + \lambda_i^\ell d_i, x_i^{\bar{\ell}}),\ \ell = 1, \ldots k. \tag{16}$$

where λ_i^ℓ, $\ell = 1, \ldots, k$, satisfy (10). Such an inequality is easily satisfied, for example, when each processor takes a desired number of steps in R^{n^ℓ} determined by any of the standard serial algorithms described in Section 2 on the function $f(x_i^\ell, x_i^{\bar{\ell}})$ starting at $(x_i^\ell + \lambda_i^\ell d_i^\ell, x_i^{\bar{\ell}})$. After these parallel steps are performed by each processor then an eventual synchronization step is needed that consists of determining x_{i+1} such that

$$f(x_{i+1}) \leq \min_{1 \leq \ell \leq k} f(y_i^\ell, x_i^{\bar{\ell}}). \tag{17}$$

We summarize these procedures as the following partially asynchronous algorithm.

Corollary 3.2 Partially Asynchronous Nonconvex PCAT1 *Theorem 3.1 holds for nonconvex f, with a resulting stationary point, if the stepsize choices (10) and synchronization procedure (11)-(12) are changed to the following:*

Partially Asynchronous Stepsize *Choose y_i^ℓ, $\ell = 1, \ldots, k$, such that for $\bar{\ell}$, the complement of ℓ in $\{1, \ldots, k,\}$:*

$$f(x_i) - f(y_i^\ell, x_i^{\bar{\ell}}) \geq \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell) \geq 0,\ \ell = 1, \ldots, k, \tag{18}$$

where μ_ℓ is a forcing function on the sequence of nonnegative real numbers $\{-\nabla_\ell f(x_i)^T d_i^\ell\}$ for bounded $\{d_i^\ell\}$, $\ell = 1, \ldots, k$.

Comment: Inequality (18) is easily implemented by satisfying (16) and (10).

Synchronization: *Find x_{i+1} such that*

$$f(x_{i+1}) \leq \min_{1 \leq \ell \leq k} f(y_i^\ell, x_i^{\bar{\ell}}). \tag{19}$$

Proof The only changes needed in the proof of Theorem 3.1 in order to apply it here are to replace δ by $\frac{1}{k}$ in (14) and to replace the string of inequalities of (13) that establish the monotonicity of

$\{f(x_i)\}$ by using (18)and (19) as follows:

$$
\begin{aligned}
f(x_i) - f(x_{i+1}) &\geq \frac{1}{k}\left[f(x_i) - f(y_i^1, x_i^2, \ldots, x_i^k)\right] + \ldots \\
&\qquad\qquad \ldots + \frac{1}{k}\left[f(x_i) - f(x_i^1, \ldots, x_i^{k-1}, y_i^k)\right] \quad \text{(By (19))} \\
&\geq \frac{1}{k}\sum_{\ell=1}^{k} \mu_\ell(-\nabla_\ell f(x_i)^T d_i^\ell).
\end{aligned}
$$

□

By combining the direction (9) and stepsize (18) choices of the Partially Asynchronous Nonconvex PCAT1 of Corollary 3.2 into a single forcing function condition (20) below, we obtain Theorem 3.2 that is a simpler and more general theorem than PCAT1 of Corollary 3.2. We omit the proof which is similar to that of Theorem 2.2.

Theorem 3.2 Partially Asynchronous Nonconvex PCAT2 *Let $f \in C^1(R^n)$ on R^n. Start with any $x_0 \in R^n$. Given x_i, stop if $\nabla f(x_i) = 0$, else determine x_{i+1} such that:*

Parallel Steps Determine y^ℓ, $\ell = 1, \ldots, k$, such that for $\bar{\ell}$ the complement of ℓ in $\{i, \ldots, k\}$:

$$f(x_i) - f(y_i^\ell, x_i^{\bar{\ell}}) \geq \sigma_\ell(\|\nabla_\ell f(x_i)\|), \ \ell = 1, \ldots, k, \tag{20}$$

where σ_ℓ is a forcing function on $\{\|\nabla_\ell f(x_i)\|\}$, for $\ell = 1, \ldots, k$.
Synchronization Step Choose x_{i+1} such that

$$f(x_{i+1}) \leq \min_{1\leq \ell \leq k} f(y_i^\ell, x_i^{\bar{\ell}}). \tag{21}$$

Either $\{x_i\}$ terminates at a stationary point $x_{\bar{i}}$, or each accumulation point $\bar{x}$ of $\{x_i\}$ is stationary.

We conclude this section with the remark that the synchronization step, in all the proposed methods in this section, can be further modified, if desired. In particular, we can search along the direction $x^i + \lambda(x^{i+1} - x^i)$, $\lambda \in R$, for a better point than x^{i+1} as the next iterate, and replace x^{i+1} by this better point. All the convergence results remain valid because of the forcing function arguments used to establish them.

4 Conclusion and Numerical Results

We have given a number of parallel versions of fundamental convergence theorems for unconstrained minimization. These basic results enable k, possible massively large, parallel processors to perform on portions of the gradient, what one processor performs on the entire gradient in a serial algorithm. The direction choices in these theorems include many of the popular directions (gradient, quasi-Newton, Newton, conjugate gradient) and stepsizes (minimization, first stationary point, interval, Armijo). Note that each processor can apply direction and stepsize choices different from those of the other processors. A synchronization step is then used to obtain a strongly convex combination of the k points obtained by the k processors for the convex case, or alternatively the best of the k points or a better point can be taken as the next iterate for the convex as well as the nonconvex case.

Numerical implementations of parallel gradient distribution algorithms have been carried out in [1, 6] on the Thinking Machines CM-5 multiprocessor. In these implementations, inexact quasi-Newton minimization was used in each parallel processor so as to satisfy (16). Each processor was allowed to take a number of steps before synchronization. The synchronization consisted of searching the affine hull of the points generated by the parallel processors as well as the current point. The problems solved in [1] consisted of real world multicategory discrimination problems, formulated as unconstrained minimization of piecewise convex quadratic functions with Lipschitz continuous gradients. Problem size varied between 70 and 140 variables. For these multicategory discrimination problems, it is most efficient to use as many parallel processors as there are categories. This happened to be 7 for the problems tested. A standard measure of efficiency for parallel algorithms is the speedup efficiency defined as

$$\text{Speedup Efficiency} = \frac{\text{Time on 1 processor}}{(\text{Time on } k \text{ processors}) * k}$$

Thus, a speedup efficiency of 100% means that the time taken by one processor is cut exactly by a factor of k, when k processors are employed. An efficiency of over 100% indicates that some of the parallel processors, that are solving smaller subproblems, have obtained very good points, or that the affine hull generated by these points spans some very good points. For the multicategory discrimination problems, speedup efficiency was between 50% and 91%. For more details see [1].

In [6], thirty unconstrained problems from the publicly available CUTE (Constrained and Unconstrained Testing Environment) [3] were tested. Among others, the parallel variable distribution algorithm version PVD0 was tested, which is equivalent to a parallel gradient distribution algorithm. Problems solved were between 100 and 1024 variables in size. These problems were solved on 2, 4, 8, 16 and 32 processors, with respective average speedup efficiencies of: 129%, 122%, 77%, 44% and 20%. These figures indicate that for problems of the size attempted, parallel gradient distribution is capable of producing a speedup, equal to or better than 44% of the number of processors used, for 16 or less processors. In order to exploit more fully a larger number of processors, larger problems need to be solved. We believe, however, that we have demonstrated that parallel gradient distribution can achieve speedups of the order of the processors employed, and hence warrant further study and testing.

References

[1] K.P. Bennett and O.L. Mangasarian. Serial and parallel multicategory dicrimination. *SIAM Journal on Optimization*, 4(4), 1994.

[2] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation.* Prentice–Hall, Inc, Englewood Cliffs, New Jersey, 1989.

[3] I. Bongartz, A.R. Conn, N. Gould, and Ph.L. Toint. CUTE: Constrained and unconstrained testing environment. Publications du Départment de Mathématique Report 93/10, Facultés Universitaires De Namur, 1993.

[4] J.W. Daniel. *The approximate minimization of functionals.* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[5] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, N.J., 1983.

[6] M.C. Ferris and O.L. Mangasarian. Parallel variable distribution. *SIAM Journal on Optimization*, 4(4), 1994.

[7] E.S. Levitin and B. T. Polyak. Constrained minimization methods. *Computational Mathematics and Mathematical Physics*, 6:1–50, 1968. Translated from Russian.

[8] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72:7–35, 1992.

[9] O.L. Mangasarian and M.V. Solodov. Serial and parallel backpropagation convergence via nonmonotone perturbed minimization. *Optimization Methods and Software*, 4(2):103–116, 1994.

[10] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, 1970.

[11] E. Polak. *Computational methods in optimization; A unified approach.* Academic Press, New York, 1971.

[12] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjugées. *Revue Francaise Informatique et Recherche Opérationelle*, 16-R1:35–43, 1969.

[13] B.T. Polyak. The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, 9(4):94–112, 1969. Translated from Russian.

[14] B.T. Polyak. *Introduction to Optimization.* Optimization Software, Inc., Publications Division, New York, 1987.

[15] P. Tseng. Dual ascent methods with strictly convex costs and linear constraints: A unified approach. *SIAM Journal on Control and Optimization*, 28:214–242, 1990.

[16] J. Warga. Minimizing certain convex functions. *Journal of SIAM on Applied Mathematics*, 11:588–593, 1963.