NONMONOTONE AND PERTURBED OPTIMIZATION

By

Mikhail V. Solodov

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN – MADISON

1995

Abstract

The primary purpose of this research is the analysis of nonmonotone optimization algorithms to which standard convergence analysis techniques do not apply. We consider methods that are inherently nonmonotone, as well as nonmonotonicity induced by data perturbations or inexact subproblem solution. One of the principal applications of our results is the analysis of gradient-type methods that process the data incrementally. The computational significance of these algorithms is well documented in the neural networks literature. Such algorithms are known to be particularly well-suited for large data sets, as well as for real-time applications. One of the most important methods of this type is the classical online backpropagation (BP) algorithm for training artificial neural networks. Neural networks constitute a large interdisciplinary area of research within the broader area of machine learning that has found applications in many branches of science and technology. However, much of the work in the area has been based on heuristic concepts and trial-and-error experimentation. This research fills some of the existing theoretical gaps. In particular, we obtain the first deterministic convergence results for the BP algorithm and its various

practically important modifications.

We also investigate error-stability properties of the generalized gradient projection method. When specialized to neural network training, our general results allow us to establish stability of BP in the presence of noise, and give its precise characterization. We also outline applications to weight perturbation training. In a classical optimization setting, some new results are derived for a perturbed generalized gradient projection method applied to convex and weakly sharp problems.

Next we develop a general approach to convergence analysis of feasible descent methods in the presence of perturbations. The important novel feature of our analysis is that perturbations need not tend to zero in the limit. In this case, standard convergence analysis techniques are not applicable, and we present a new approach. It is shown that a certain ε-approximate solution can be obtained, where ε depends on the level of perturbations linearly. Applications to the gradient projection, proximal minimization and extragradient algorithms are described.

We also consider a practical generalization of the parallel variable distribution algorithm of Ferris and Mangasarian. In particular, our generalization is twofold : we propose an asynchronous algorithm, and allow inexact subproblem solution. We show that the generalized method retains all the attractive properties of the original method and yet is more practical. We also derive some stronger convergence results for algorithms of this class.

Acknowledgements

I wish to express my greatest thanks to my research advisor, Professor Olvi Mangasarian. His advice, encouragement and support have been invaluable. His enthusiasm and scientific vision have been and will always remain an inspiration for me. I feel very fortunate to have worked under his guidance.

I wish to thank Professors Stephen Robinson and John Strikwerda for serving on my committee and Robert Meyer and Amos Ron for serving as readers. I also thank them, Michael Ferris, Carl de Boor and Jude Shavlik for their courses that I have taken over the years and for many informal discussions.

I thank my family for their love and constant support. The guidance and encouragement from my mother, Valentina Solodova, and my father, Vladimir Solodov, were instrumental in my getting an excellent initial education and pursuing a scientific career. I express my deepest gratitude for all the sacrifices that they made for me over the years.

I also would like to thank the faculty of the Department of Computational Mathematics and Cybernetics at the Moscow State University who contributed greatly to my education. Special thanks go to my undergraduate advisor Professor Sergei Zavriev who introduced me to the field of Optimization and was instrumental in my coming to Madison. I also thank my ex-wife Liba Brent whose support over the last few years was very important to me, as well as all of my many friends, where ever they are.

This research was partially supported by Air Force Office of Scientific Research Grant F49620-94-1-0036 and National Science Foundation Grant CCR-9322479.

Contents

Chapter 1

Convergence of Incremental Gradient-Type Methods

A general convergence theorem is proposed for a family of serial and parallel nonmonotone unconstrained minimization methods with perturbations [42]. A principal application of the theorem is to establish convergence of incremental gradient-type methods. Of special interest is online backpropagation (BP), the classical algorithm for training artificial neural networks. Under certain natural assumptions, such as divergence of the sum of the stepsizes and convergence of the sum of their squares, it is shown that every accumulation point of the BP iterates is a stationary point of the error function associated with the given set of training examples. The results presented cover serial and parallel BP, as well as modified BP with a momentum term.

1.1 Incremental Gradient Methods And Neural Network Training

We consider the problem of minimizing a summation of a finite number of continuously differentiable functions over the n-dimensional real space

$$\min_{x \in \Re^n} f(x) := \sum_{j=1}^{K} f_j(x), \tag{1.1.1}$$

where the integer parameter K is typically large. Note that problem (1.1.1) can be viewed as an extension of the standard optimization problem which can be obtained by setting $K = 1$. Optimization problems of this form naturally arise in many practical applications. Least norm minimization is one such example.

One of the basic iterative methods for solving (1.1.1) is the gradient method given by

$$x^{i+1} = x^i - \eta_i \nabla f(x^i), \quad i = 0, 1, \ldots ,$$

where η_i is a positive stepsize. In the neural networks literature, this is often referred to as *batch* iteration. In applications where K is large, a batch iteration may be very costly, and the standard gradient method is essentially cost prohibitive. Naturally, in that case more sophisticated techniques, such as conjugate gradient or quasi-Newton methods, are also inapplicable. Unfortunately, this often seems to be the case in machine learning. For many practical neural network applications, standard optimization methods require storage and computational cost which can become unmanageable even for a moderate network size, provided the training set is large enough [59]. For problems of this kind, **incremental methods** are known to be more cost effective and often less likely to get stuck at poor local minima or stationary points of which there are many [37]. Such methods do not wait to process the entire set of functions $f_j(\cdot),\ j = 1, \ldots, K$ before updating the current iterate. Every iteration of the incremental gradient method is a step in the direction of negative gradient of a partial objective function. The method can be expressed in the following cyclical form :

$$
\begin{aligned}
& x^{i+1} = x^i - \eta_i \nabla f_1(x^i), \\
& \vdots \\
& x^{i+t} = x^{i+t-1} - \eta_{i+t-1} \nabla f_t(x^{i+t-1}), \\
& \vdots \\
& x^{i+K+1} = x^{i+K} - \eta_{i+K} \nabla f_K(x^{i+K}), \\
& \text{set } i := i + K + 1, \text{ repeat.}
\end{aligned}
$$

In the neural networks literature, methods of this type are usually referred to as **online** BP and often erroneously stated to be descent methods, which they are not. The computational significance of these methods is well documented. They are known to be particularly well suited for large data sets (K is large). Another attractive feature of online approach is that it is incremental and can be used in real time operation. These properties are of particular importance for optimal control [1] and artificial intelligence [52] applications. A more detailed comparison of the online and batch approaches to training neural networks is given below (see page 7).

A neural network can be thought of as a network representation of a certain nonlinear map between an input space and an output space. A principal goal of constructing this map is to correctly discriminate between the elements of two finite (typically disjoint) sets in the input space $\Re^m$. In this setting, the output space is the binary set $\{0, 1\}$. A neural network consists of a set of weighted arcs and a set of nodes with thresholds (see Figure 1). A node takes as input $\zeta = w^\top \xi$ and produces its output by applying an *activation* function to this weighted input. Motivated by the human neuron, in most theoretical models the following step activation function is used

$$step(\zeta - \theta) = \begin{cases} 0 & \text{if } \zeta \leq \theta, \\ 1 & \text{if } \zeta > \theta, \end{cases}$$

where θ is the threshold of the unit. Thus a unit is activated when its input exceeds its threshold. The neural network depicted in Figure 1, has an input vector $\xi^j \in \Re^m$, one layer of h hidden units with threshold values $\theta^i \in \Re$, incoming arc weights $w^i \in \Re^m$, outgoing arc weights $v^i \in \Re$, $i = 1, \ldots, h$, and an output unit with a threshold $\tau \in \Re$ and output $y(\xi^j) \in \{0, 1\}$.

Thus, neural networks are parametrized by a set of weights and thresholds. The task of a training scheme for a neural network is to find a set of weights and thresholds that makes the network perform the desired mapping. This problem can be formalized as follows [36].

Neural Network Training Problem

Given two finite disjoint sets in $\Re^m$, $\mathcal{A}_0$ and $\mathcal{A}_1$, determine a positive integer

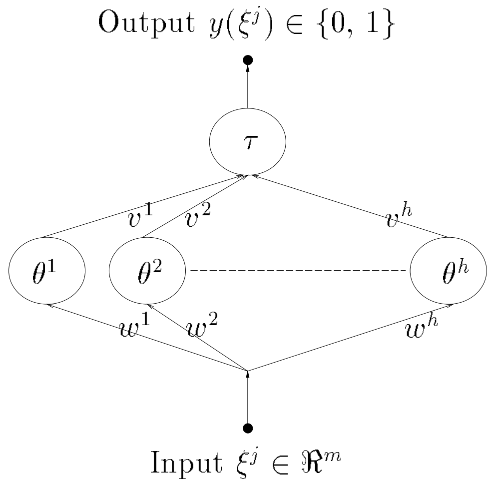

Figure 1: A feedforward neural network with a single layer of h hidden units

h, $w^i \in \Re^m$, $\theta^i \in \Re$, $v^i \in \Re$, $i = 1, \ldots, h$, and $\tau \in \Re$, such that the output $y(\xi^j)$ of the neural network satisfies $y(\xi^j) = 0$ for $\xi^j \in \mathcal{A}_0$ and $y(\xi^j) = 1$ for $\xi^j \in \mathcal{A}_1$, $j = 1, \ldots, K$.

The fact that this problem is solvable if we employ a sufficient number of hidden units (h is sufficiently large), essentially follows from the Kolmogorov Approximation Theorem (see [22]). Note however, that choosing too large an h may lead to overtraining and memorizing the training set without the ability to generalize to unseen data [24]. The choice of h is in general a rather difficult task in itself, and it is beyond the scope of this work. From now on, we assume that h is chosen and fixed for each particular problem.

To apply optimization theory to the training of an artificial neural network, we think of the Neural Network Training Problem as minimization of the following least squares error function :

$$\min_{x \in X \subset \Re^n} f(x, \alpha) := \sum_{j=1}^{K} f_j(w, \theta, v, \tau, \alpha), \tag{1.1.2}$$

$$f_j(w, \theta, v, \tau, \alpha) := \left(s \left(\sum_{i=1}^{h} s(\xi^j w^i - \theta^i, \alpha) v^i - \tau, \alpha \right) - t^j \right)^2 , \tag{1.1.3}$$

where

h = fixed integer number of hidden units

K = fixed integer number of given training samples ξ^j in $\Re^m$

$t^j = 0$ or 1 target value for $y(\xi^j),\ j = 1, \ldots, K$

τ = real number, threshold of output unit

v^i = real numbers, weights of outgoing arcs from hidden units, $i = 1, \ldots, h$

θ^i = real numbers, thresholds of hidden units, $i = 1, \ldots, h$

w^i = m-vector weights of incoming arcs to hidden units, $i = 1, \ldots, h$

ξ^j = given m-dimensional vector samples, $j = 1, \ldots, K$

$s(\zeta, \alpha) = \dfrac{1}{1 + e^{-\alpha\zeta}}$, $\alpha > 0$.

Here α is the smoothing parameter of the *sigmoid* approximation $s(\zeta, \alpha)$ of the discontinuous step function $step(\zeta) = 1$ if $\zeta > 0$, else 0. The set X is typically either $\Re^n$ or a set of simple box-constraints.

One of the most widely used methods for training neural networks is the backpropagation algorithm (BP) [57]. Every iteration of online BP is a step in the direction of negative gradient of a partial error function associated with a single training example (e.g. $f_j(\cdot, \alpha)$ in (1.1.2)). In its simplest form, online BP can be written as the following iterative process :

$$x^{i+1} = x^i - \eta_i \nabla f_{m(i)}(x^i, \alpha), \quad i = 0, 1, \ldots ,$$

where η_i is a positive stepsize (learning rate), and $m(\cdot)$ is a single-valued map from the positive integers to the set $\{1, \ldots , K\}$. For simplicity, we assume that for every span of K iterations, the map $m(\cdot)$ generates each of the indices $1, \ldots , K$ exactly once. For now, we also assume that the smoothing parameter α is fixed. It is clear that there is no guarantee that a step of BP will decrease the full objective function $f(\cdot, \alpha)$, which is the sum of the errors for *all* the training examples . A single iteration of BP may, in fact, increase rather than decrease the objective function $f(\cdot, \alpha)$ which we are trying to minimize. This difficulty makes convergence analysis of BP a challenging problem that has currently attracted the interest of many researchers [42, 32, 70].

To further justify our interest in the online (incremental) methods we make the following observations [20] :

(i) For many learning systems adaptation to on-line stream of training samples is required. If all the training examples are not available before the training starts, the online procedure is essentially the only way to go.

(ii) The online method is usually faster and more effective than batch methods for large scale classification problems, especially for data sets with redundant information. By redundancy we mean that contributions of gradients of many of the partial error functions to the total gradient are very similar. Therefore, waiting to compute all these contributions before updating the weights could be a waste of time.

(iii) The online procedure naturally introduces some randomness (noise) that may help the iterates escape from a stationary point or a "poor" local minimum.

(iv) The online method allows for simpler (thus more reliable) hardware on-chip implementation.

Of course, for some problems, the batch approach may be more effective. This is particularly true if application of conjugate gradient, quasi-Newton, or other more sophisticated optimization techniques is feasible. The latter, however, is not the case for real time on-chip implementations.

We briefly describe our notation now. The usual inner product of two vectors $x \in \Re^n$, $y \in \Re^n$ is denoted by $\langle x, y \rangle$. The Euclidean 2-norm of $x \in \Re^n$ is given by $\|x\|^2 = \langle x, x \rangle$. For a real-valued matrix A of any dimension, $A^\top$ denotes its transpose. For a differentiable function $f : \Re^n \to \Re$, ∇f will denote the n-dimensional vector of partial derivatives with respect to x, and $\nabla_l f$ will denote the n_l-dimensional vector of partial derivatives with respect to $x_l \in \Re^{n_l}$, $n_l \leq n$. If a function $f(\cdot)$ is continuously differentiable on $\Re^n$, we shall write $f(\cdot) \in C^1(\Re^n)$. If $f(\cdot)$ has Lipschitz continuous partial derivatives on $\Re^n$ with some constant $L > 0$, that is

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\| \;\; \forall\, x, y \in \Re^n,$$

we write $f(\cdot) \in C^1_L(\Re^n)$. $\Re_+$ will denote the nonnegative real line, that is $\Re_+ := \{x \in \Re \mid x \geq 0\}$. For two nonnegative scalar functions $s_1 : \Re_+ \to \Re_+$ and $s_2 : \Re_+ \to \Re_+$, we say that $s_1 = O(s_2)$ if there exists a positive constant c such that $\limsup_{t\to\infty} \frac{s_1(t)}{s_2(t)} = c$. By R-linear convergence and Q-linear convergence, we mean linear convergence in the root sense and in the quotient sense, respectively, as defined in [47].

1.2 Convergence Of A Class Of Nonmonotone Algorithms

We start with a convergent nonmonotone algorithm theorem for the solution of the unconstrained minimization problem

$$\min_{x \in \Re^n} f(x), \tag{1.2.1}$$

where $f : \Re^n \to \Re$ is a continuously differentiable function from the n-dimensional real space $\Re^n$ to the real numbers $\Re$. Our result is much in the spirit of [33], except for the key difference of nonmonotonicity. This generalization allows the proposed results to apply to a wider class of algorithms including backpropagation. Theorem 1.2.1 below proved to be very useful and has since been used in [2, 26] for convergence analysis of other incremental algorithms.

We first define a forcing function.

Definition 1.2.1 *A continuous function* $\sigma : \Re_+ \to \Re_+$ *such that* $\sigma(0) = 0, \sigma(t) > 0$ *for* $t > 0$, *and such that* $t^i \geq 0$ *and* $\{\sigma(t^i)\} \to 0$ *imply that* $\{t^i\} \to 0$, *is said to be a forcing function.*

Some typical examples of forcing functions are ct, ct^2 for some $c > 0$.

We now state a classical lemma ([51],p.6) that will be used later, as well as another lemma (a slight modification of [51],p.44) used in the proof of Theorem 1.2.1.

Lemma 1.2.1 *Let* $f(\cdot) \in C^1_L(\Re^n)$, *then*

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in \Re^n.$$

Lemma 1.2.2 *Let* $\{a^i\}$ *and* $\{\epsilon^i\}$ *be two sequences of real numbers such that* $\epsilon^i \geq 0$, $\sum_{i=0}^{\infty} \epsilon^i < \infty$, *and* $a^{i+1} \leq a^i + \epsilon^i$ *for* $i = 0, 1, \ldots.$ *It follows that either the sequence* $\{a^i\}$ *is unbounded below, or it converges.*

We are now ready to state and prove our first result. The first part of the proof is fairly standard, while some novel arguments are needed to establish the second assertion.

Theorem 1.2.1 *Let $f(\cdot) \in C^1(\Re^n)$. Start with any $x^0 \in \Re^n$. Having x^i stop if $\nabla f(x^i) = 0$, else compute $x^{i+1} = x^i + \eta_i d^i$ according to a direction d^i and stepsize η_i chosen as follows*

Direction d^i **:**

$$- \langle \nabla f(x^i), d^i \rangle \geq \sigma(\|\nabla f(x^i)\|) - \lambda_i, \tag{1.2.2}$$

where $\lambda_i \geq 0$ and $\sigma(\cdot)$ is a forcing function .

Stepsize η_i **:**

$$f(x^i) - f(x^{i+1}) \geq -\eta_i \langle \nabla f(x^i), d^i \rangle - \nu_i, \quad \eta_i > 0, \nu_i \geq 0. \tag{1.2.3}$$

Let the following conditions hold

$$\sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \lambda_i \eta_i < \infty, \quad \sum_{i=0}^{\infty} \nu_i < \infty. \tag{1.2.4}$$

Then either the sequence $\{f(x^i)\}$ is unbounded below, or it converges and $\inf_i \|\nabla f(x^i)\| = 0$. If, in addition, $f(\cdot) \in C_L^1(\Re^n)$ and $\|d^i\| \leq c, \ \forall\ i, \quad c > 0$, it follows that $\{\nabla f(x^i)\} \to 0$, and for each accumulation point point $\bar{x}$ of the sequence $\{x^i\}$, $\nabla f(\bar{x}) = 0$.

Proof. If $\nabla f(x^i) = 0$ for some i, then the algorithm terminates at a stationary point. Suppose now that it does not terminate.

Combining (1.2.2) and (1.2.3) we have

$$f(x^i) - f(x^{i+1}) \geq \eta_i \sigma(\|\nabla f(x^i)\|) - \lambda_i \eta_i - \nu_i \ . \tag{1.2.5}$$

Hence

$$f(x^{i+1}) \leq f(x^i) + \lambda_i \eta_i + \nu_i \ .$$

By (1.2.4) and Lemma 1.2.2, either $\{f(x^i)\} \to -\infty$, or $\{f(x^i)\}$ converges. From now on we assume that the latter holds. Then for some $\bar{f} \in \Re$, it follows that $f(x^i) \geq \bar{f}$ for all i.

Applying (1.2.5) to the first summation below we obtain

$$\begin{aligned} f(x^0) - \bar{f} &\geq f(x^0) - f(x^i) \\ &= \sum_{j=0}^{i-1} (f(x^j) - f(x^{j+1})) \\ &\geq \sum_{j=0}^{i-1} \eta_j \sigma(\|\nabla f(x^j)\|) - \sum_{j=0}^{i-1} (\lambda_j \eta_j + \nu_j) \\ &\geq \inf_{0 \leq j \leq i-1} \sigma(\|\nabla f(x^j)\|) \sum_{j=0}^{i-1} \eta_j - \sum_{j=0}^{i-1} \lambda_j \eta_j - \sum_{j=0}^{i-1} \nu_j \ . \end{aligned} \tag{1.2.6}$$

By letting $i \to \infty$ we obtain

$$f(x^0) - \bar{f} \geq \inf_{j \geq 0} \sigma(\|\nabla f(x^j)\|) \sum_{j=0}^{\infty} \eta_j - \sum_{j=0}^{\infty} \lambda_j \eta_j - \sum_{j=0}^{\infty} \nu_j \ . \tag{1.2.7}$$

Since the left-hand-side and the last two terms of the right-hand-side in (1.2.7) are finite numbers, it follows from the divergence of $\sum_{j=0}^{\infty} \eta_j$ that $\inf_j \sigma(\|\nabla f(x^j)\|) = 0$. By Definition 1.2.1 of the forcing function we immediately have that

$$\inf_i \|\nabla f(x^i)\| = 0. \tag{1.2.8}$$

Now assume that $f(\cdot) \in C_L^1(\Re^n)$ and $\|d^i\| \leq c, \ \forall i, \ c > 0$. Suppose the sequence $\{\nabla f(x^i)\}$ does not converge to zero. Then there exists some $\epsilon > 0$ and some increasing sequence of integers $\{i_l\}$ such that $\|\nabla f(x^{i_l})\| \geq \epsilon$ for all l. On the other hand, (1.2.8) guarantees that for every l there exists some $j > i_l$ such that $\|\nabla f(x^j)\| \leq \frac{\epsilon}{2}$. For each l let $j(l)$ denote the least integer which satisfies these conditions. By the triangle inequality, the fact that $f(\cdot) \in C_L^1(\Re^n)$ and (1.2.4), we have

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \|\nabla f(x^{i_l})\| - \|\nabla f(x^{j(l)})\| \\
&\leq \|\nabla f(x^{i_l}) - \nabla f(x^{j(l)})\| \\
&\leq L\|x^{i_l} - x^{j(l)}\| \\
&\leq L \sum_{t=i_l}^{j(l)-1} \eta_t \|d^t\| \\
&\leq Lc \sum_{t=i_l}^{j(l)-1} \eta_t \ .
\end{aligned}$$

Hence

$$\sum_{t=i_l}^{j(l)-1} \eta_t \geq \frac{\epsilon}{2Lc} = \bar{c} > 0. \tag{1.2.9}$$

By making use of (1.2.5) and (1.2.9), we have

$$\begin{aligned}
f(x^{i_l}) - f(x^{j(l)}) &\geq \sum_{t=i_l}^{j(l)-1} \eta_t \sigma(\|\nabla f(x^t)\|) - \sum_{t=i_l}^{j(l)-1} (\lambda_t \eta_t + \nu_t) \\
&\geq \bar{c} \inf_{i_l \leq t \leq j(l)-1} \sigma(\|\nabla f(x^t)\|) - \sum_{t=i_l}^{\infty} (\lambda_t \eta_t + \nu_t) \ .
\end{aligned}$$

Since the sequence $\{f(x^i)\}$ converges and the last summation above converges to zero as $l \to \infty$, it follows that

$$\lim_{l\to\infty} \inf_{i_l \le t \le j(l)-1} \sigma(\|\nabla f(x^t)\|) = 0. \tag{1.2.10}$$

However, by the choice of i_l and $j(l)$, $\|\nabla f(x^t)\| \ge \frac{\epsilon}{2}$, $\forall t : i_l \le t < j(l)$. This contradicts (1.2.10) since $\sigma(\cdot)$ is a forcing function. Hence the assumption that $\nabla f(x^i)$ does not converge to zero is invalid. Taking into account continuity of the gradient of $f(\cdot)$, we conclude that if $\bar{x}$ is an accumulation point of $\{x^i\}$, then $\nabla f(\bar{x}) = 0$. The proof is complete. ■

Remark 1.2.1 *Assumptions (1.2.2), (1.2.3) and (1.2.4) can be combined into the following simpler and more general condition, where θ_i replaces $\lambda_i \eta_i + \nu_i$:*

$$f(x^i) - f(x^{i+1}) \ge \eta_i \sigma(\|\nabla f(x^i)\|) - \theta_i \ ,$$

$$\sum_{i=0}^{\infty} \eta_i = \infty, \ \ \sum_{i=0}^{\infty} \theta_i < \infty.$$

These new conditions also guarantee that the assertions of Theorem 1.2.1 hold. However, we have chosen to state Theorem 1.2.1 in a direction – stepsize form because it is easier to implement. See [33] for specific instances of directions d^i and stepsize η_i choices without perturbation terms.

We now show that Theorem 1.2.1 can be applied to the analysis of the perturbed gradient-type methods. It is important to point out that the assumptions (1.2.11) below of Corollary 1.2.1 of Lipschitz continuity and boundedness of $\nabla f(\cdot)$ can be all satisfied in the context of BP, the convergence of which is established in Section 1.3. In addition, the BP error function is guaranteed to be bounded from below.

Corollary 1.2.1 *Let*

$$f(\cdot) \in C^1_L(\Re^n), \quad \|\nabla f(x)\| \le M \quad \forall x \in \Re^n, \quad \textit{for some } M > 0. \tag{1.2.11}$$

Start with any $x^0 \in \Re^n$*. Having* x^i*, stop if* $\nabla f(x^i) = 0$*, else compute*

$$x^{i+1} = x^i + \eta_i d^i, \tag{1.2.12}$$

where

$$d^i = -\nabla f(x^i) + e^i \tag{1.2.13}$$

for some $e^i \in \Re^n$*,* $\eta_i \in \Re$*,* $\eta_i > 0$ *and such that*

$$\sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \eta_i^2 < \infty, \quad \sum_{i=0}^{\infty} \eta_i \|e^i\| < \infty, \quad \|e^i\| \le \gamma \;\; \forall i, \quad \gamma > 0. \tag{1.2.14}$$

It follows that either $\{f(x^i)\} \to -\infty$*, or* $\{f(x^i)\}$ *converges,* $\{\nabla f(x^i)\} \to 0$ *and for each accumulation point* $\bar{x}$ *of the sequence* $\{x^i\}$*,* $\nabla f(\bar{x}) = 0$*.*

Proof. It suffices to show that conditions (1.2.2)–(1.2.4) of Theorem 1.2.1 are satisfied. We first note that, by (1.2.11), (1.2.13) and (1.2.14), $\|d^i\| \le M + \gamma$ for all i.

By the Cauchy-Schwartz inequality, (1.2.11) and (1.2.12), we have with $\sigma(s) = s^2$,

$$\begin{aligned} -\langle \nabla f(x^i), d^i \rangle &= \|\nabla f(x^i)\|^2 - \langle \nabla f(x^i), e^i \rangle \\ &\geq \sigma(\|\nabla f(x^i)\|) - \|\nabla f(x^i)\| \|e^i\| \\ &\geq \sigma(\|\nabla f(x^i)\|) - M\|e^i\|. \end{aligned} \tag{1.2.15}$$

By Lemma 1.2.1, (1.2.12) and (1.2.13), it follows

$$\begin{aligned} f(x^i) - f(x^{i+1}) &\geq -\langle \nabla f(x^i), x^{i+1} - x^i \rangle - \frac{L}{2}\|x^{i+1} - x^i\|^2 \\ &= -\eta_i \langle \nabla f(x^i), d^i \rangle - \frac{L}{2}\eta_i^2 \|d^i\|^2 \\ &\geq -\eta_i \langle \nabla f(x^i), d^i \rangle - \frac{L}{2}\eta_i^2 (M + \gamma)^2. \end{aligned} \tag{1.2.16}$$

Relations (1.2.15), (1.2.16) and (1.2.14) establish the assumptions (1.2.2)–(1.2.4) of Theorem 1.2.1 with $\lambda_i = M\|e^i\|, \;\; \nu_i = \frac{L}{2}(M + \gamma)^2 \eta_i^2$.

The proof is complete. ∎

Remark 1.2.2 *Under appropriate assumptions, other well known direction choices, such as conjugate and quasi-Newton directions [51] can also be perturbed similarly as in Corollary 1.2.1.*

Remark 1.2.3 *Similar to [33], a parallel version of Theorem 1.2.1 can be established where portions of the gradient are distributed among the processors. However, having in mind the analysis of the BP algorithm, we shall instead here concentrate on parallel distribution of the additive objective function.*

We thus consider an extension of problem (1.2.2) to the case when the objection function may be given by the sum of a finite number of functions :

$$\min_{x \in \Re^n} f(x) := \sum_{j=1}^{K} f_j(x) \tag{1.2.17}$$

Suppose that we have p parallel processors, $p \geq 1$. Let J_l be a partition of $\{1, \ldots, K\}$ such that $J_l \subseteq \{1, \ldots, K\}$, $\cup_{l=1}^{p} J_l = \{1, \ldots, K\},;$ $J_{l_1} \cap J_{l_2} = \emptyset$ for $l_1 \neq l_2$. Let K_l be the number of elements in J_l. We define the function $f^l(\cdot)$ associated with J_l as follows

$$f^l(x) := \sum_{j \in J_l} f_j(x) \tag{1.2.18}$$

With this definition we have

$$f(x) = \sum_{l=1}^{p} f^l(x) \tag{1.2.19}$$

We are now ready to state and prove a parallel version of Corollary 1.2.1.

Theorem 1.2.2 *Let each $f_j(\cdot)$, $j = 1, \ldots, K$, satisfy the assumptions (1.2.11) of Corollary 1.2.1. Start with any $x^0 \in \Re^n$. Having x^i, stop if $x^i = x^{i-1}$. Else compute x^{i+1} as follows:*

($\bullet$) Parallelization : *For each processor $l \in \{1, \ldots, p\}$ compute*

$$y_l^{i+1} = x^i + \eta_i d_l^i, \tag{1.2.20}$$

where

$$d_l^i = -\nabla f^l(x^i) + e_l^i, \quad \eta_i > 0. \tag{1.2.21}$$

(•) Synchronization : *Let*

$$x^{i+1} = \frac{1}{p}\sum_{l=1}^{p} y_l^{i+1}. \tag{1.2.22}$$

If for some $\gamma > 0$

$$\sum_{i=0}^{\infty}\eta_i = \infty, \quad \sum_{i=0}^{\infty}\eta_i^2 < \infty, \quad \sum_{i=0}^{\infty}\eta_i\|e_l^i\| < \infty, \quad \|e_l^i\| \leq \gamma, \ \forall\, i, \quad l = 1, \ldots, p \tag{1.2.23}$$

then all the conclusions of Corollary 1.2.1 hold.

Proof. We shall establish assumptions (1.2.12)– (1.2.14) of Corollary 1.2.1.

By (1.2.19) and (1.2.20)–(1.2.22), we have

$$\begin{aligned} x^{i+1} - x^i &= \frac{1}{p}\sum_{l=1}^{p} y_l^{i+1} - x^i \\ &= \frac{1}{p}\sum_{l=1}^{p}(x^i + \eta_i d_l^i) - x^i \\ &= \frac{1}{p}\sum_{l=1}^{p}\eta_i d_l^i \\ &= \frac{\eta_i}{p}\sum_{l=1}^{p}(-\nabla f^l(x^i) + e_l^i) \\ &= \frac{\eta_i}{p}\left(-\nabla f(x^i) + \sum_{l=1}^{p} e_l^i\right) \end{aligned}$$

Now, in view of (1.2.23), Corollary 1.2.1 applies with $e^i = \sum_{l=1}^{p} e_l^i$, and the proof is complete. ■

Remark 1.2.4 *Theorem 1.2.2 can be easily generalized so that each processor takes an arbitrary but finite number of steps before any synchronization is made.*

The changes needed to extend Theorem 1.2.2 to these asynchronous methods are straightforward, and are thus omitted.

Remark 1.2.5 *The synchronization step in Theorem 1.2.2 can be modified so that other linear combination of y_l^{i+1}, $l = 1, \ldots, p$ with positive (equal) weights is taken. However, this complicates the proof somewhat.*

1.3 Convergence Of The Backpropagation Algorithm

We now turn our attention to the classical BP algorithm for training feedforward artificial neural networks with one layer of hidden units [57, 24]. The number of hidden units is assumed to be fixed.

Suppose we have K training examples and p processors with $K \geq 1$ and $p \geq 1$ (typically, K is much bigger than p). In a manner similar to that of Section 1.2 we consider a partition of the set $\{1, \ldots, K\}$ into the subsets J_l, $l = 1, \ldots, p$, so that each example is assigned to at least one processor. For empirical study of this kind of parallelization see [49, 14]. Note that if examples are assigned to more than one processor, a different (weighted) error function may be generated, which nevertheless measures the error of the same training problem. The variables of the problem here are the weights associated with the arcs of the neural network and the thresholds of the hidden and output units. The objective is to minimize a certain error function (see Section 1.1) which we shall write as

$$\min_{x \in \Re^n} f(x) := \sum_{l=1}^{p} f^l(x) = \sum_{l=1}^{p} \sum_{j \in J_l} f_j(x).$$

We note that this function is the sum of individual error functions each of which is associated with a single training example. Each component $f_j(\cdot)$ of the objective function is a squared composition of the sigmoid and linear functions, and therefore satisfies the assumptions (1.2.11) on any bounded set. In this section we assume the smoothing parameter α to be fixed, and skip the dependence of $f(\cdot)$ on α in our notation.

Each iteration of the *serial online* BP algorithm consists of a step in the direction of negative gradient of an error function associated with a single training example. In the *parallel* BP each processor performs one (or more) cycles of serial BP on its set of training examples. Then a synchronization step is performed that consists of averaging the iterates computed by all the p processors. Empirical evaluation of parallel BP and numerical tests can be found in [49, 14].

Below we state a parallel BP algorithm with an added *momentum term* which consists of the difference between the current and previous iterates. For simplicity and in a similar manner to the method of conjugate gradients [51] we reset this term to zero periodically (see Algorithm 1.3.1). It has been observed that introduction of momentum term usually leads to faster convergence and adds stability to problems with noisy data [24].

We now summarize and describe our notation for stating and establishing convergence of the parallel BP algorithm with a momentum term :

$\boldsymbol{i = 1, 2, \ldots}$: Index number of major iterations of BP, each of which consists of going through the entire set of error functions $f_1(x), \ldots, f_K(x)$. This is achieved serially or in parallel by p processors with processor l handling the error function $f^l(x),\ l = 1, \ldots, p$.

$\boldsymbol{j = 1, \ldots, K_l}$: Index of minor iterations performed by parallel processor $l, \quad l = 1, \ldots, p$. Each minor iteration j consists of a step in the direction of negative gradient $-\nabla f^l_{m(j)}(z_l^{i,j})$ and a momentum step, where $m(j)$ is an element of the permuted set J_l. Note that in general, the map $m(\cdot)$ depends on the index i and processor l. For simplicity, we skip this dependence in our notation. Recall that K_l is the number of elements in the set J_l.

$\boldsymbol{x^i}$: Iterate in $\Re^n$ of major iteration $i = 1, 2, \ldots$.

$\boldsymbol{z_l^{i,j}}$: Iterate in $\Re^n$ of minor iteration $j = 1, \ldots, K_l$, within major iteration $i = 1, 2, \ldots$, computed by processor $l = 1, \ldots, p$.

By $\boldsymbol{\eta_i}$ we shall denote the *learning rate* (the coefficient multiplying the gradient), and by $\boldsymbol{\alpha_i}$ the *momentum rate* (the coefficient multiplying the momentum term). For simplicity we shall assume that the learning and momentum rates remain fixed within each major iteration. Table 1 gives a flowchart of the parallel BP algorithm.

We are now ready to state and prove convergence of the parallel BP algorithm.

Algorithm 1.3.1 Parallel BP with Momentum Term.
Start with any $x^0 \in \Re^n$. *Having* x^i, *stop if* $\nabla f(x^i) = 0$, *else compute* x^{i+1} *as*

follows :

(•) **Parallelization :** *for each processor* $l \in \{1, \ldots, p\}$ *do*

$$z_l^{i,j+1} = z_l^{i,j} - \eta_i \nabla f_{m(j)}^l(z_l^{i,j}) + \alpha_i \Delta z_l^{i,j}, \quad j = 1, \ldots, K_l, \tag{1.3.1}$$

$$\textit{where } z_l^{i,1} = x^i, \quad 0 < \eta_i < 1, \quad 0 \leq \alpha_i < 1.$$

$$\Delta z_l^{i,j} = \begin{cases} 0 & \textit{if } j = 1 \\ z_l^{i,j} - z_l^{i,j-1} & \textit{otherwise} \end{cases} \tag{1.3.2}$$

(•) **Synchronization :**

$$x^{i+1} = \frac{1}{p} \sum_{l=1}^{p} z_l^{i,K_l+1} \tag{1.3.3}$$

We note that for $p = 1$, Algorithm 1.3.1 becomes the serial BP, while the choice of $\alpha_i = 0$ reduces it to the simple BP. Note that since each parallel processor can use the same program for its computations, load balancing can be easily achieved. For the sake of simplicity, we consider the synchronization step of averaging the weights obtained by all the parallel processors. It is possible to take other (equally) weighted linear combinations.

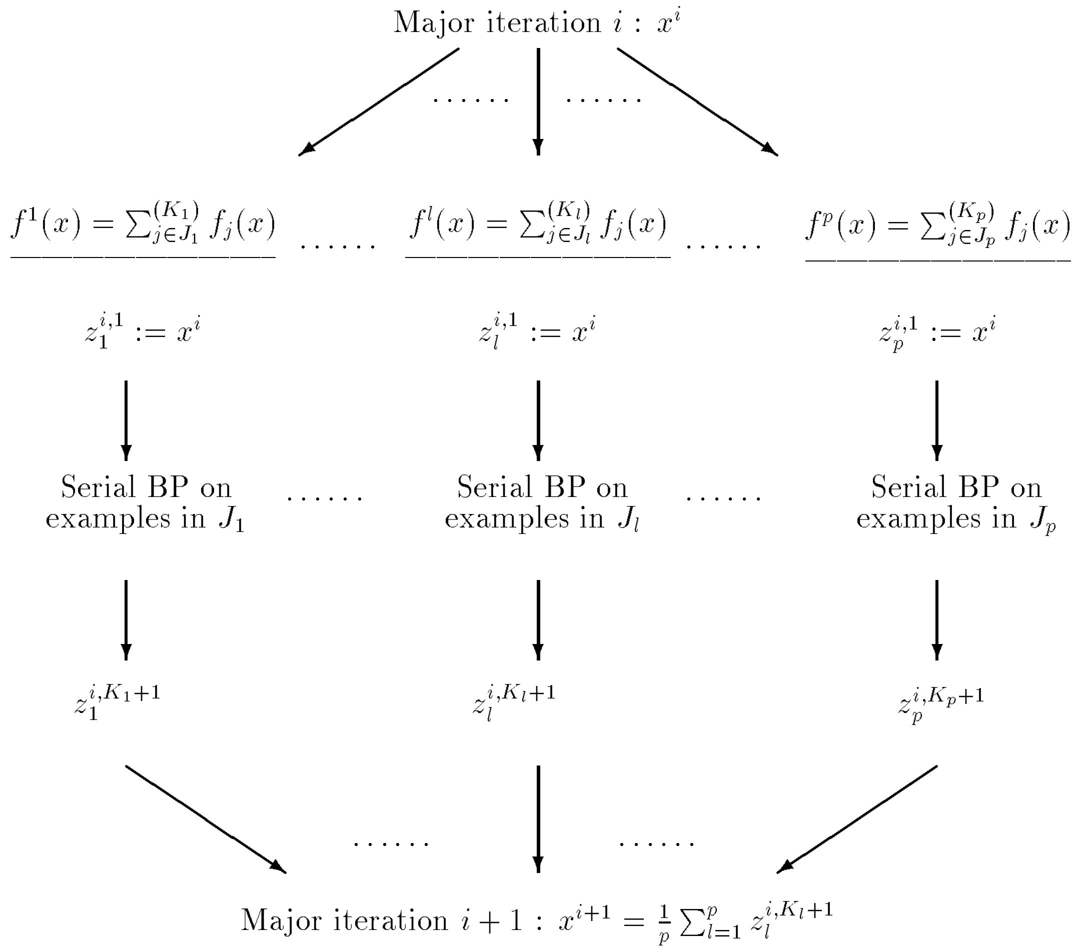

Table 1. Flowchart of the Parallel BP

Below we give the first deterministic convergence proof for the parallel and serial BP algorithm. In [70] it is proven that the sequence of weights generated by the serial BP either converges to a point that is almost surely stationary or it diverges. In contrast, our approach is deterministic. Our proof which is based on the results of Section 1.2, covers both serial and parallel cases as well as the computationally important methods with a momentum term.

We are now ready to apply the analysis of Section 1.2 to backpropagation.

Theorem 1.3.1 *Let $S \subset \Re^n$ be any bounded set. If the learning and momentum rates are chosen so that*

$$\sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \eta_i^2 < \infty, \quad \sum_{i=0}^{\infty} \alpha_i \eta_i < \infty, \tag{1.3.4}$$

then for any sequence $\{x^i\} \subset S$ generated by the BP Algorithm 1.3.1 it follows that $\{f(x^i)\}$ converges, $\{\nabla f(x^i)\} \to 0$, and for each accumulation point $\bar{x}$ of the sequence $\{x^i\}$, $\nabla f(\bar{x}) = 0$.

Proof. We shall show that the assumptions of Theorem 1.2.2 are satisfied. First note that the error function is smooth and thus its gradient satisfies assumptions (1.2.11) on a bounded set S. Also $f(\cdot)$ is nonnegative, hence it is bounded below. Using (1.3.1) and (1.3.2), for any cycle i, any processor l, and any j such that $2 \leq j \leq K_l + 1$ we obtain

$$\begin{aligned}
z_l^{i,j} - x^i &= z_l^{i,j} - z_l^{i,1} \\
&= \sum_{t=1}^{j-1} (z_l^{i,t+1} - z_l^{i,t}) \\
&= \sum_{t=1}^{j-1} (-\eta_i \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i \Delta z_l^{i,t}) \\
&= -\eta_i \sum_{t=1}^{j-1} \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i (z_l^{i,j-1} - x^i) \qquad (1.3.5) \\
&= -\eta_i \sum_{t=1}^{j-1} \nabla f_{m(t)}^l (z_l^{i,t}) - \eta_i \sum_{s=1}^{j-2} \left(\alpha_i^{j-1-s} \sum_{t=1}^{s} \nabla f_{m(t)}^l (z_l^{i,t}) \right), (1.3.6)
\end{aligned}$$

where (1.3.6) is obtained by repeated use of (1.3.5) with j replaced by $j-1,\ j-2,\dots,2$. By (1.3.5) and (1.2.18), for $j = K_l + 1$ we have

$$\begin{aligned} z_l^{i,K_l+1} - x^i &= -\eta_i \sum_{t=1}^{K_l} \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i (z_l^{i,K_l} - x^i) \\ &= -\eta_i (\nabla f^l(x^i) + a_l^i + \frac{\alpha_i}{\eta_i} b_l^i), \end{aligned} \tag{1.3.7}$$

where

$$a_l^i = \sum_{t=2}^{K_l} (\nabla f_{m(t)}^l (z_l^{i,t}) - \nabla f_{m(t)}^l (x^i)), \tag{1.3.8}$$

and

$$b_l^i = x^i - z_l^{i,K_l}. \tag{1.3.9}$$

Let

$$e_l^i = -a_l^i - \frac{\alpha_i}{\eta_i} b_l^i. \tag{1.3.10}$$

Now, in view of Theorem 1.2.2, assumptions (1.3.4), and (1.3.7), all we have to do is to verify that

$$\sum_{i=0}^{\infty} \eta_i \|e_l^i\| < \infty, \ \ \|e_l^i\| \le \gamma, \ \ \gamma > 0,\ l = 1, \dots, p. \tag{1.3.11}$$

By (1.3.8), (1.3.6), (1.2.11), the triangle inequality and $\alpha_i \le 1$ it follows

$$\begin{aligned} \|a_l^i\| &\le \sum_{t=2}^{K_l} \|\nabla f_{m(t)}^l (z_l^{i,t}) - \nabla f_{m(t)}^l (x^i)\| \\ &\le L \sum_{t=2}^{K_l} \|z_l^{i,t} - x^i\| \\ &\le L \sum_{t=2}^{K_l} \left(\eta_i \sum_{r=1}^{t-1} \|\nabla f_{m(r)}^l (z_l^{i,r})\| + \eta_i \sum_{s=1}^{t-2} \left(\alpha_i^{t-1-s} \sum_{r=1}^{s} \|\nabla f_{m(r)}^l (z_l^{i,r})\| \right) \right) \\ &\le L\eta_i (K_l^2 M + K_l^3 M) \\ &= c_1 \eta_i. \end{aligned} \tag{1.3.12}$$

Similarly, by (1.3.9), (1.3.6), (1.2.11), the triangle inequality and $\alpha_i \leq 1$, we have

$$\begin{aligned} \|b_l^i\| &= \|z_l^{i,K_l} - x^i\| \\ &\leq \eta_i \left(\sum_{t=1}^{K_l-1} \|\nabla f_{m(t)}^l(z_l^{i,t})\| + \sum_{s=1}^{K_l-2} \left(\alpha_i^{K_l-1-s} \sum_{t=1}^{s} \|\nabla f_{m(t)}^l(z_l^{i,t})\| \right) \right) \\ &\leq \eta_i \left(\sum_{t=1}^{K_l-1} M + \sum_{s=1}^{K_l-2} MK_l \right) \\ &\leq \eta_i(MK_l + MK_l^2) \\ &= c_2\eta_i. \end{aligned} \tag{1.3.13}$$

By (1.3.10), (1.3.12), (1.3.13), and the triangle inequality, we obtain

$$\|e_l^i\| \leq c_1\eta_i + c_2\alpha_i$$

The latter combined with (1.3.4) implies (1.3.11), and the proof is complete. ∎

Remark 1.3.1 *In general, a constant stepsize, no matter how small, does not guarantee convergence of online BP iterates to stationary points of the objective function, even in the weak sense of Theorem 1.3.1.*

This can be verified by considering a one-dimensional "two-piece" strongly convex quadratic function

$$f(x) = f_1(x) + f_2(x) := \frac{1}{2}x^2 + \frac{1}{2}(x-1)^2.$$

Note that the unique stationary point of $f(\cdot)$ is $\bar{x} = 1/2$. Let $x^0 = 0$. Suppose the learning rate is fixed at some value $\eta \in (0,1)$. For any i,

$$x^{2i+1} = x^{2i} - \eta\nabla f_1(x^{2i}) = (1-\eta)x^{2i},$$

and

$$x^{2i+2} = x^{2i+1} - \eta \nabla f_2(x^{2i+1}) = (1-\eta)x^{2i+1} + \eta.$$

Combining the above two equations, we obtain the following linear recurrence relation

$$x^{2i+2} = (1-\eta)^2 x^{2i} + \eta.$$

Taking into account that $x^0 = 0$ and $0 < \eta < 1$ it can be verified that

$$\lim_{i\to\infty} x^{2i} = \frac{1-\eta}{2-\eta}.$$

Similarly,

$$\lim_{i\to\infty} x^{2i+1} = \frac{1}{2-\eta}.$$

The two accumulation points are distinct (and different from $\bar{x}$!) for any fixed η. Note however, that they both tend to $\bar{x}$ in the limit as $\eta \to 0$.

Remark 1.3.2 *We note that the learning rates rules that satisfy (1.3.4) were used in practice [10], and are known as search-then-converge strategy. In the first phase of learning, called the "search phase", the learning rate is almost constant, or it decreases slowly. In the second phase of learning, called the "converge phase", it decreases to zero.*

In particular, two possible rules for the learning rate have been suggested [10], [9] :

$$\eta_i = \eta_0 \frac{1}{1+\frac{i}{i_0}}$$

and

$$\eta_i = \eta_0 \frac{1 + \frac{ci}{\eta_0 i_0}}{1 + \frac{ci}{\eta_0 i_0} + i_0(\frac{i}{i_0})^2}$$

where $\eta_0 > 0$, $c > 0$, $i_0 >> 1$ are appropriately chosen parameters. Note that for $i << i_0$, the learning rate $\eta_i \cong \eta_0$, and for $i >> i_0$ the learning rate decreases proportional to $1/i$. Hence these rules satisfy the conditions of Theorem 1.3.1.

Remark 1.3.3 *The boundedness assumption in Theorem 1.3.1 was made in order to ensure that $f(\cdot)$ has Lipschitz continuous and bounded gradient. Since the principal result of the theorem deals with accumulation points, boundedness of the iterates is needed in order to ensure the existence of such accumulation points.*

There are a number of ways to ensure that the sequence of iterates produced by BP be bounded, such as the following.

In [16] a regularization term consisting of the squared 2-norm of x is added to the error function so that the modified objective function has bounded level sets :

$$\min_{x \in \Re^n} f(x) := \sum_{j=1}^{K} f_j(x) + c\|x\|^2,$$

where $c > 0$ is a (small) "penalty" parameter. This, in fact, corresponds to the *weight decay* training [21, 69]. Weight decay is a useful approach since it tends to generate simpler networks by minimizing nonzero arc connections. Simpler networks often possess better generalization properties. All of our results apply with merely redefining the objective function of the problem.

We could also consider the constrained problem

$$\min_{x \in X} f(x) := \sum_{j=1}^{K} f_j(x),$$

where X is typically a box in $\Re^n$ [32]. Then a simple projection onto X ensures that the iterates are bounded. Only some technical changes are needed to apply our analysis to the constrained version of BP [32] which is the same as Algorithm 1.3.1, except that the synchronization step concludes with a projection operation

$$x^{i+1} = \left[\frac{1}{p} \sum_{l=1}^{p} z_l^{i,K_l+1} \right]^+,$$

where $[\cdot]^+$ denotes the orthogonal projection onto X.

1.4 Concluding Remarks

A general theorem for the nonmonotone convergence of a family of unconstrained optimization methods has been presented. It was established that the serial or parallel backpropagation algorithm with or without a momentum term for training feedforward artificial neural networks with one layer of hidden units can be viewed as a *deterministic* perturbed gradient-type method. Each accumulation point of the sequence of weights generated by BP is shown to be a stationary point of the error function associated with the given set of training examples. The results of this chapter are equally applicable to feedforward neural networks with more than one layer of hidden units. Other generalizations are possible.

Chapter 2

Generalized Gradient Projection Methods in The Presence of Perturbations

We investigate convergence properties of the generalized gradient projection algorithm in the presence of perturbations [62]. It is shown that the iterates of the method are attracted, in a certain sense, to an ε-stationary set of the problem, where ε depends on the magnitude of the perturbations. Characterization of the attraction sets for the iterates is given in the general (nonsmooth and nonconvex) case. In the convex case, convergence to an ϵ-optimal set is established. The results are further strengthened for weakly sharp and strongly convex problems. Convergence of the parallel incremental algorithm for minimizing an additive objective function is established. We also present applications to stability analysis of algorithms for training artificial neural networks.

2.1 Introduction

We consider the following general optimization problem

$$\min_{x \in X} f(x), \tag{2.1.1}$$

where X is a convex compact set in $\Re^n$. Let B denote the closed unit ball in $\Re^n$, that is $B := \{x \in \Re^n \mid \|x\| \leq 1\}$. For the objective function, we assume that there exists $\tau \in (0, +\infty]$ such that $f : (X + \tau B) \to \Re$ is at least Lipschitz continuous on $X + \tau B$ and regular (in the sense of Clarke, [6]).

Let X_{opt} and X_s denote the optimal and stationary sets of problem (2.1.1) respectively, that is

$$X_{opt} := \{x \in X \mid f(x) = \min_{y \in X} f(y)\}$$

and

$$X_s := \{x \in X \mid 0 \in \partial f(x) + N_X(x)\},$$

where $\partial f(x)$ is the set of all generalized gradients (in the sense of Clarke, [6]) of $f(\cdot)$ at x, and $N_X(x) \subset \Re^n$ is the normal cone to the set X at the point $x \in X$ (see [54]) :

$$N_X(x) = \{y \in \Re^n \mid \langle y, z - x\rangle \leq 0 \quad \forall z \in X\}.$$

The following notions will play an important role in our analysis. Let $\varepsilon : X \to \Re_+$ be any nonnegative upper semicontinuous function. We define the

$\varepsilon(\cdot)$ -stationary set of the problem (2.1.1) as follows :

$$X_s(\varepsilon(\cdot)) := \{x \in X \mid 0 \in \partial f(x) + N_X(x) + \varepsilon(x)B\}.$$

Clearly, $X_s = X_s(0)$. For any nonnegative upper semicontinuous function $\epsilon : X \to \Re_+$, the $\epsilon(\cdot)$-optimal set of (2.1.1) is defined by

$$X_{opt}(\epsilon(\cdot)) := \{x \in X \mid f(x) \leq \min_{y \in X} f(y) + \epsilon(x)\}.$$

Obviously, $X_{opt}(0) = X_{opt}$. In the convex case, the sets $X_s(\varepsilon(\cdot))$ and $X_{opt}(\epsilon(\cdot))$ are related in a certain way (see Lemma 2.4.2). In that case, many of our general results can be considerably strengthened (see Section 2.4).

In this chapter we establish stability properties of the generalized gradient projection method

$$x^{new} := [x - \eta(g(x) + \delta(x))]^+, \quad g(x) \in \partial f(x), \quad \eta \to 0,$$

where $\delta(\cdot)$ represents perturbations (noise) and $[\cdot]^+$ denotes the orthogonal projection map onto X. We also study various important modifications of this basic algorithm, including incremental methods (see Algorithms 2.3.1, 2.3.2, 2.4.1). We point out that the condition of decaying stepsize ($\eta \to 0$) is indispensable in the general nonsmooth case [46], as well as in the case of (smooth) incremental methods (see Remark 1.3.1). In this chapter we show that the iterates of the algorithm are, in a certain sense, attracted to an $\varepsilon(\cdot)$-stationary set of the problem (Theorem 2.3.1). We give a precise characterization of $\varepsilon(\cdot)$ in terms of asymptotic behavior of perturbations. Our analysis is based on the novel technique presented in [76]. This approach allows us to deal with essentially perturbed problems (i.e. problems with nonvanishing noise : $\delta(x^i) \not\to 0$ as $i \to \infty$), as well as analyze algorithms that are inherently nonmonotone, e.g. incremental methods (see Algorithm 2.3.1).

For every $x \in X$ we define the following nonnegative scalar function $r : X \to \Re_+$

$$r(x) := \{\min \|h\| \mid h \in \partial f(x) + N_X(x)\}. \tag{2.1.2}$$

It is clear that $r(\cdot)$ is an optimality function for problem (2.1.1) in the sense that

$$r(x) \begin{cases} = 0 & \text{if } x \in X_s \\ > 0 & \text{otherwise} \end{cases}$$

From the definitions of $X_s(\varepsilon(\cdot))$ and $r(x)$, we immediately obtain the following key relation

$$X_s(\varepsilon(\cdot)) = \{x \in X \mid r(x) \leq \varepsilon(x)\}. \tag{2.1.3}$$

Let $\mathcal{F}(\cdot,\cdot) : \mathcal{N} \times X \to \mathcal{M}(\Re^m)$ be a point-to-set mapping (or a multifunction), where $\mathcal{M}(C)$ denotes the set of all subsets of a set C, and $\mathcal{N}$ denotes the nonnegative integers. We define the *upper topological limit* of $\mathcal{F}(\cdot,\cdot)$ at $x \in \Re^n$ by

$$\mathop{\overline{\mathrm{lt}}}_{\substack{x'(\in X)\to x \\ i\to\infty}} \mathcal{F}(i,x') := \left\{ y \in \Re^m \;\middle|\; \begin{array}{l} \text{there exist sequences } \{x'_i\},\ \{m_i\} \text{ and } \{y_i\} \\ \text{such that } y_i \in \mathcal{F}(m_i, x'_i),\ i = 1,2,\dots, \\ \{x'_i\} \to x,\ \{m_i\} \to \infty \text{ as } i \to \infty, \\ \text{and } y = \lim_{i\to\infty} y_i \end{array} \right\}$$

In particular, for a bounded sequence $\{x_i\} \subset X$, $\bar{\mathrm{lt}}_{i\to\infty}\{x_i\}$ denotes the set of all accumulation points of $\{x_i\}$. We say that a sequence $\{x_i\}$ converges *into* set C, if $\bar{\mathrm{lt}}_{i\to\infty}\{x_i\} \subset C$.

Note that under our assumptions,

$$\bar{\mathrm{lt}}_{x'(\in X)\to x} N_X(x') = N_X(x) \ \ \forall x \in X. \tag{2.1.4}$$

Of particular interest for us will be an extension of problem (2.1.1) to the case when the objective function $f(\cdot)$ is given by a summation of a finite number of functions $f_j(\cdot, \alpha_0)$, $j = 1, \ldots, K$. Note that we further allow the dependence of f_j on a parameter. We thus consider the problem

$$\min_{x\in X} f(x, \alpha_0) := \sum_{j=1}^{K} f_j(x, \alpha_0). \tag{2.1.5}$$

For every $j \in \{1, \ldots, K\}$, the function $f_j : (X + \tau B) \times A \to \Re$ involves a parameter $\alpha \in A \subset \Re$ that may vary during the optimization process. We assume that the set A is bounded. Problems of the form (2.1.5) arise, for example, in least-norm minimization, neural networks applications, and approximation theory. Among some important practical applications that involve parameters in the objective function, we note the adaptive smoothing techniques [45], and the neural network training [57, 41, 36]. We assume that each function $f_j(\cdot, \alpha)$ is Lipschitz continuous with modulus $L > 0$ and regular on an open neighborhood of $X + \tau B$ for every $\alpha \in A$. We also assume that the map $\partial f_j(\cdot, \cdot)$ is upper semicontinuous. That is, for all $j \in \{1, \ldots, K\}$,

$$\bar{\mathrm{lt}}_{\substack{x'(\in X)\to x \\ \alpha(\in A)\to\alpha_0}} \partial f_j(x', \alpha) \subset \partial f_j(x, \alpha_0) \ \ \forall x \in X, \tag{2.1.6}$$

where $\partial f_j(x, \alpha)$ denotes the set of all generalized gradients of $f_j(\cdot, \alpha)$ at $x \in X$.

The rest of this chapter is organized as follows. In Section 2.2 we outline the Generalized Lyapunov Direct Method for stability analysis. In Section 2.3 we establish convergence properties of the generalized gradient projection method and its modifications in the presence of data perturbations. Section 2.4 contains the results that are strengthened for the case of weakly sharp and convex problems. In Section 2.5, we relate our work to neural network training.

One more word about our notation. By conv C we shall denote the convex hull of a set C, and by int C its interior.

2.2 Generalized Lyapunov Direct Method

In this section we outline the novel convergence analysis technique that was first proposed in [76] (albeit in a slightly different form). This technique can be viewed as a generalization of the Lyapunov Direct Method for convergence analysis of nonlinear iterative processes. The Lyapunov Direct Method has proved to be a powerful tool for stability analysis of both continuous and discrete time processes [56, 72, 50, 51]. Roughly speaking, this approach reduces analysis of stability properties of a process to the analysis of local improvement of this process with respect to some scalar criterion $V(\cdot)$ (usually called the *Lyapunov function*). In the classical approach, $V(\cdot)$ monotonically decreases on the iterates of the process (some typical choices for $V(\cdot)$ are : the objective function being minimized, the norm of its gradient, the distance to the solution set (see [51])).

The key difference of the presented technique is that we relax the monotonicity requirement. We thus refer to $V(\cdot)$ as a **pseudo-Lyapunov function**. This generalization makes our approach applicable to a wider class of algorithms, including methods with perturbations.

We now state the Generalized Lyapunov Direct Method. The convergence (attraction) properties of the process are expressed in terms of a pseudo-Lyapunov function $V(\cdot)$. For each specific algorithm, those properties allow further interpretation depending on the choice of $V(\cdot)$ for this algorithm.

We consider the following general iterative process

$$x^{i+1} \in x^i - \eta_i G(i, x^i) - \xi^i, \quad i = 0, 1, \dots, \quad x^0 \in X', \quad \xi^i \in \Re^n, \tag{2.2.1}$$

$$\eta_i \to 0, \quad \sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \xi^i \text{ is (component-wise) convergent,} \tag{2.2.2}$$

where $G(\cdot, \cdot) : \mathcal{N} \times X' \to \mathcal{M}(X')$, and X' is an open set in $\Re^n$. In applications, ξ^i usually corresponds to (random) noise. We further make the natural boundedness assumption

$$\sup_{x \in X'} \limsup_{\substack{x' \to x \\ i \to \infty}} \sup_{y \in G(i, x')} \|y\| < \infty .$$

Thus the upper topological limit of $G(\cdot, \cdot)$, denoted by

$$G_0(x) := \operatorname*{\bar{lt}}_{\substack{x' \to x \\ i \to \infty}} G(i, x'),$$

is bounded and upper semicontinuous on a neighborhood of any compact set $X \subset X'$.

We assume that there exists a compact set $X \subset X'$ which contains all the accumulation points of the iterates generated by (2.2.1)-(2.2.2), that is

$$\overline{\mathrm{lt}}_{i\to\infty}\{x^i\} \subset X. \tag{2.2.3}$$

Suppose a pseudo-Lyapunov function $V(\cdot)$ is chosen. Let $V(\cdot)$ be Lipschitz continuous and regular on a neighborhood of X. For the pseudo-Lyapunov function $V(\cdot)$, the set X, and the map $G_0(\cdot)$, we define the following set which is crucial for our analysis :

$$\mathcal{A}_0 := \{x \in X \mid \max_{h\in H(x)} \min_{g\in G_0(x)} \langle h, g\rangle \leq 0\}, \tag{2.2.4}$$

where

$$H(x) = \mathrm{conv}\{\partial V(x) \cup N_X(x)\}.$$

Roughly speaking, the set $\mathcal{A}_0$ is comprised of all the points in X for which $-G_0(x)$ does not contain feasible directions that are of descent for the pseudo-Lyapunov function $V(\cdot)$.

The following result shows that the sequences generated by (2.2.1)- (2.2.2) and satisfying (2.2.3) are, in a certain sense, attracted to the components of the set $\mathcal{A}_0$. We first have to introduce the notion of $V(\cdot)$-connected components of $\mathcal{A}_0$ (recall that $\mathcal{A}_0$ is compact). We say that a set $C \subset \Re^n$ is $V(\cdot)$-connected, if the set $V(C) = \{v \in \Re \mid \exists x \in C,\ \ v = V(x)\} \subset \Re$ is connected. Let $\{\mathcal{A}^\gamma\},\ \ \gamma \in$

Γ be the (unique) decomposition of $\mathcal{A}_0$ into $V(\cdot)$-connected components (see [73]), that is

$$\mathcal{A}_0 = \cup_{\gamma\in\Gamma}\mathcal{A}^\gamma, \;\; \mathcal{A}^{\gamma'} \neq \mathcal{A}^{\gamma''} \;\text{ for } \gamma' \neq \gamma'', \;\; \gamma', \gamma'' \in \Gamma.$$

The following theorem will play a central role in the subsequent analysis.

Theorem 2.2.1 *[76] For every sequence $\{x^i\}$ generated by the process (2.2.1)-(2.2.2), and satisfying (2.2.3), there exists a $\gamma \in \Gamma$ such that the following properties hold :*

$$\underset{i\to\infty}{\bar{\mathrm{lt}}}\, V(x^i) = V\left(\underset{i\to\infty}{\bar{\mathrm{lt}}}\{x^i\} \cap \mathcal{A}^\gamma\right),$$

and every subsequence $\{x^{i_m}\}$ of $\{x^i\}$ satisfying

$$\lim_{m\to\infty} V(x^{i_m}) = \liminf_{i\to\infty} V(x^i) \quad \mathit{or} \quad \lim_{m\to\infty} V(x^{i_m}) = \limsup_{i\to\infty} V(x^i)$$

converges into $\mathcal{A}^\gamma$.

Corollary 2.2.1 *[76] Let the set $V(\mathcal{A}_0)$ be nowhere dense in $\Re$. Then every sequence $\{x^i\}$ generated by the process (2.2.1)-(2.2.2), and satisfying (2.2.3), converges into a connected component of $\mathcal{A}_0$.*

2.3 Convergence Properties of a Parallel Generalized Gradient Projection Algorithm in the Presence of Perturbations

In this section we consider the problem (2.1.5) of minimizing an additive parametric objective function. We first describe our notation for stating and establishing convergence of the parallel perturbed generalized gradient projection method (GGPM) for solving (2.1.5) and its modifications. The type of parallelization considered here is primarily motivated by incremental gradient methods, particularly neural network training (see Chapter 1). Empirical evaluation of parallel BP and numerical tests can be found in [49, 14]. Another related work on parallel computing is [68]. We first consider the most general case. Our results can be then specialized by removing parallelism and/or considering the standard (nonadditive) objective function.

The notation is similar to that for Algorithm 1.3.1.

$\boldsymbol{i = 1, 2, \ldots}$: Index number of major iterations of GGPM, each of which consists of going through the entire set of functions $f_1(x, \alpha_i), \ldots, f_K(x, \alpha_i)$. This is achieved serially or in parallel by p processors with processor l handling at the i-th iteration the functions $f_j(x, \alpha_i)$, $j \in J_l$. Recall that $\alpha_i \in A$ is the (smoothing) parameter, and $\lim_{i \to \infty} \alpha_i = \alpha_0$. For simplicity, we assume that the sets J_l, $l = 1, \ldots, p$ are ordered as follows

$$\begin{aligned} J_1 = &\quad \{1, \ldots, K_1\}, \\ J_2 = &\quad \{K_1 + 1, \ldots, K_1 + K_2\}, \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ J_p = &\quad \{K_1 + \cdots + K_{p-1} + 1, \ldots, K\}, \end{aligned}$$

i.e.

$$J_l = \{\bar{K}_l + 1, \ldots, \bar{K}_l + K_l\},\ l = 1, \ldots, p,$$

where

$$\bar{K}_l = \sum_{t=1}^{l-1} K_t,\ l = 2, \ldots, p,\ \bar{K}_1 = 0.$$

$\boldsymbol{j = 1, \ldots, K_l}$: Index of minor iterations performed by parallel processor l, $l = 1, \ldots, p$. Each minor iteration j consists of a step in the direction of a negative generalized gradient $-\tilde{g}_l^{i,j}$ of the function $f_{\bar{K}_l + j}(\cdot, \alpha_i)$ at $z_l^{i,j}$ which is calculated with some error $\delta_l^{i,j}$:

$$\begin{aligned} &\tilde{g}_l^{i,j} = g_l^{i,j} + \delta_l^{i,j}, \\ &g_l^{i,j} \in \partial f_{\bar{K}_l + j}(z_l^{i,j}, \alpha_i), \\ &\delta_l^{i,j} = \delta_{\bar{K}_l + j}(z_l^{i,j}, \alpha_i, i). \end{aligned}$$

The function $\delta_j(z, \alpha, i)$ denotes perturbation of the generalized gradient of $f_j(\cdot, \alpha)$ at the point $z \in X + \tau B$ at the i-th major iteration of the algorithm. With respect to those perturbations we make the following mild boundedness assumption :

$$\sum_{j=1}^{K} \sup_i \sup_{z \in X + \tau B} \sup_{\alpha \in A} \|\delta_j(z, \alpha, i)\| < \infty.$$

$\boldsymbol{x^i}$: Iterate in $\Re^n$ of major iteration $i = 1, 2, \ldots$.

$\boldsymbol{z_l^{i,j}}$: Iterate in $\Re^n$ of minor iteration $j = 1, \ldots, K_l$, within major iteration $i = 1, 2, \ldots$, computed by processor $l = 1, \ldots, p$.

By $\boldsymbol{\eta_i}$ we shall denote the stepsize, i.e. the coefficient multiplying the generalized gradients at the i-th major iteration. For simplicity we shall assume that η_i remains fixed within each major iteration. We consider the process with stepsizes decreasing subject to the following condition

$$\eta_i > 0,\ i = 0, 1, \ldots, \quad \eta_i \to 0,\ \sum_{i=0}^{\infty} \eta_i = \infty. \tag{2.3.1}$$

Note that under our assumptions, there exists a constant $M > 0$ such that

$$\|y\| \leq M \qquad \forall y \in \partial f_j(x, \alpha_i) + \delta_j(x, \alpha_i, i), \tag{2.3.2}$$
$$\forall x \in X + \tau B,\ \ i = 0, 1, \ldots,\ \ j = 1, \ldots, K.$$

We are now ready to state and prove convergence properties of the parallel GGPM in the presence of perturbations.

Algorithm 2.3.1 (Parallel GGPM) *Start with any* $x^0 \in X$. *Having* x^i, *compute* x^{i+1} *as follows :*

(•) Parallelization : *for each processor* $l \in \{1, \ldots, p\}$ *do*

$$z_l^{i,j+1} = z_l^{i,j} - \eta_i \tilde{g}_l^{i,j},\ \ j = 1, \ldots, K_l, \tag{2.3.3}$$

where $z_l^{i,1} = x^i$.

(•) Synchronization :

$$x^{i+1} = \left[x^i + \sum_{l=1}^{p} (z_l^{i,K_l+1} - x^i) \right]^+ \tag{2.3.4}$$

Note that for $K = 1,\ p = 1$, Algorithm 2.3.1 becomes a standard (perturbed) generalized gradient projection method, while $K \geq 2,\ p = 1$ gives an incremental backpropagation-type method. Thus the framework considered here is very general.

There are two sources of nonmonotonicity that come into play in Algorithm 2.3.1. First of all, each direction is associated with a generalized gradient of a partial objective function $f_j(\cdot, \alpha_i)$. Thus even if this direction is that of descent for $f_j(\cdot, \alpha_i)$, there is no guarantee that it is also of descent for the full objective function $f(\cdot, \alpha_0)$ given by (2.1.5) (also note a possible difference in the parameter value). The other source of nonmonotonicity is induced by perturbations of the generalized gradients.

We first verify that the (minor) iterates remain within the set $X + \tau B$ and hence are well defined.

Lemma 2.3.1 *If the stepsizes are chosen so that*

$$\eta_i \leq \frac{\tau}{M \max_l K_l} \tag{2.3.5}$$

then

$$z_l^{i,j+1} \in X + \tau B, \quad i = 0, 1, \dots, \quad j = 1, \dots, K_l, \ \ l \in \{1, \dots, p\}.$$

Proof. Consider any $i \geq 1$ and any $l \in \{1, \dots, p\}$. By the synchronization step (2.3.4),

$$z_l^{i,1} = x^i \in X \subset X + \tau B.$$

For any $j = 1, \ldots, K_l$,

$$\begin{aligned}
\|z_l^{i,j+1} - x^i\| &= \|\sum_{t=1}^{j}(z_l^{i,t+1} - z_l^{i,t})\| \\
&\leq \eta_i \sum_{t=1}^{j} \|\tilde{g}_l^{i,t}\| \\
&\leq \eta_i (j-1) M \\
&\leq \eta_i M \max_l K_l \\
&\leq \tau,
\end{aligned}$$

where the first inequality follows from the Cauchy-Schwartz inequality, and the second from (2.3.2). The result follows. ∎

From now on, we assume that the stepsizes satisfy both (2.3.1) and (2.3.5).

To analyze the influence of computational errors $\delta_l^{i,j}$ on the convergence properties of the algorithm, we need to estimate the level of perturbations in the limit. We say that $\varepsilon(x)$ is the **exact asymptotic level of perturbations** at a point $x \in X$, if

$$\varepsilon(x) = \limsup_{\substack{z_j(\in X+\tau B)\to x(\in X) \\ i\to\infty}} \|\sum_{j=1}^{K} \delta_j(z_j, \alpha_i, i)\|. \tag{2.3.6}$$

It is easy to see that the function $\varepsilon(\cdot) : X \to \Re_+$ is upper semicontinuous.

The following simple lemma proves to be very useful.

Lemma 2.3.2 *For every $x \in X$, $g \in \Re^n$, and $\eta > 0$ the following property* *holds :*

$$y = [x - \eta g]^+ \Longrightarrow \exists h \in N_X(y),\ \|h\| \leq \|g\|,\ \ y = x - \eta(g+h). \tag{2.3.7}$$

Proof. Let $y = [x - \eta g]^+$. By properties of the projection operator ([51],p.121),

$$\langle x - \eta g - y, z - y\rangle \leq 0 \quad \forall z \in X.$$

By the definition of the normal cone, the latter is equivalent to

$$x - \eta g - y \in N_X(y).$$

Hence there exists an $s \in N_X(y)$ such that $x - \eta g - y = s$. Denoting $h = \frac{1}{\eta} s \in N_X(y)$, we have that $y = x - \eta(g+h)$. Finally,

$$\begin{aligned} \|g\|^2 &= \frac{1}{\eta^2}\|x-y\|^2 - \frac{1}{\eta}\langle x - y, h\rangle + \|h\|^2 \\ &\geq \frac{1}{\eta^2}\|x-y\|^2 + \|h\|^2 \\ &\geq \|h\|^2, \end{aligned}$$

where the first inequality follows from $\eta > 0$, $x \in X$ and the definition of the normal cone. ∎

Using Lemma 2.3.2, we can re-write the synchronization step (2.3.4) as

$$x^{i+1} = x^i + \sum_{l=1}^{p}(z_l^{i,K_l+1} - x^i) + h, \quad h \in N_X(x^{i+1}). \tag{2.3.8}$$

By (2.3.3),

$$z_l^{i,K_l+1} = x^i - \eta_i \sum_{j \in J_l} (g_l^{i,j} + \delta_l^{i,j}). \tag{2.3.9}$$

Combining (2.3.8) and (2.3.9), we obtain

$$x^{i+1} = x^i - \eta_i \sum_{l=1}^{p} \sum_{j \in J_l} (g_l^{i,j} + \delta_l^{i,j}) + h,$$

where

$$h \in N_X(x^{i+1}), \quad g_l^{i,j} \in \partial f_{\bar{K}_{l-1}+j}(z_l^{i,j+1}).$$

Using these relations, we next introduce a map $G(\cdot,\cdot) : \mathcal{N} \times X \to \Re^n$ such that every sequence $\{x^i\}$ generated by Algorithm 2.3.1 is a trajectory of the iterative process

$$x^{i+1} \in x^i - \eta_i G(i, x^i), \quad i = 0, 1, \dots, \quad x^0 \in X.$$

We will refer to such $G(\cdot,\cdot)$ as the *characteristic mapping* of the algorithm. For Algorithm 2.3.1, we have

$$G(i,x) = \left\{ v \in \Re^n \;\middle|\; \begin{array}{l} v = \sum_{l=1}^{p} \sum_{j \in J_l} (g_l^j + \delta_l^j) + h, \text{ where} \\ h \in N_X(y), \ \|h\| \le MK, \text{ and} \\ y = x + \sum_{l=1}^{p} (z_l^{K_l+1} - x) + h, \\ z_l^{j+1} = z_l^j - \eta_i (g_l^j + \delta_l^j), \quad z_l^1 = x \\ g_l^j \in \partial f_{\bar{K}_{l-1}+j}(z_l^j, \alpha_i), \\ \delta_l^j = \delta_{\bar{K}_l+j}(z_l^j, \alpha_i, i), \\ j = 1, \dots, K_l, \quad l = 1, \dots, p. \end{array} \right\} \tag{2.3.10}$$

Obviously, by (2.3.2),

$$\|v\| \le 2MK \quad \forall v \in G(i,x), \quad i = 0, 1, \dots, \quad \forall x \in X.$$

Hence the map $G(\cdot,\cdot)$ is bounded, and so is its upper topological limit. We are now ready to apply the Generalized Lyapunov Direct Method of Section 2.2 to establish the properties of Algorithm 2.3.1.

We first have to estimate the upper topological limit of $G(\cdot,\cdot)$. Because $\eta_i \to 0$ (see (2.3.1)), as $x' \to x,\ i \to \infty$, we have $z_l^j \to x,\ j = 1, \ldots, K_l + 1,\ l = 1, \ldots, p$ and $y \to x$. Therefore, by the upper semicontinuity of $\partial f(\cdot,\cdot)$ and $N_X(\cdot)$ (see (2.1.4) and (2.1.6)) and the definition (2.3.6) of $\varepsilon(\cdot)$, we have

$$G_0(x) := \mathop{\bar{\mathrm{lt}}}_{\substack{x' \to x \\ i \to \infty}} G(i, x') \subset \partial f(x) + N_X(x) + \varepsilon(x) B. \tag{2.3.11}$$

Consider the decomposition of of the set $X_s(\varepsilon(\cdot))$ into the union of $f(\cdot)$-connected components

$$X_s(\varepsilon(\cdot)) = \cup_{\gamma \in \Gamma} X_s(\varepsilon(\cdot))^\gamma$$

(see Section 2.2). Our main result is the following

Theorem 2.3.1 *For every sequence $\{x^i\}$ generated by Algorithm 2.3.1, there exists $\gamma \in \Gamma$ such that the following properties hold :*

$$\mathop{\bar{\mathrm{lt}}}_{i \to \infty} f(x^i) = f\left(\mathop{\bar{\mathrm{lt}}}_{i \to \infty} \{x^i\} \cap X_s(\varepsilon(\cdot))^\gamma \right),$$

and every subsequence $\{x^{i_m}\}$ of $\{x^i\}$ satisfying

$$\lim_{m \to \infty} f(x^{i_m}) = \liminf_{i \to \infty} f(x^i) \quad or \quad \lim_{m \to \infty} f(x^{i_m}) = \limsup_{i \to \infty} f(x^i) \tag{2.3.12}$$

converges into $X_s(\varepsilon(\cdot))^\gamma$.

In particular, if $\varepsilon(\cdot) \equiv 0$ and the set $f(X_s)$ is nowhere dense in $\Re$, then every sequence $\{x^i\}$ generated by Algorithm 2.3.1 converges to a connected component of X_s.

Proof. We choose

$$V(x) := f(x),$$

where $f(x)$ is given by (2.1.5), as the pseudo-Lyapunov function of the iterative process. Following the approach outlined in Section 2.2, we introduce the set

$$\mathcal{A}_0 := \{x \in X \mid \max_{h \in H(x)} \min_{g \in G_0(x)} \langle h, g\rangle \leq 0\},$$

where $H(x) := \mathrm{conv}\{\partial f(x) \cup N_X(x)\}$. Our proof is by virtue of showing that

$$\mathcal{A}_0 \subset X_s(\varepsilon(\cdot)),$$

and then applying Theorem 2.2.1 and Corollary 2.2.1.

For every $x \in X$ we define

$$h_0(x) = \arg\min\{\|h\| \mid h \in \partial f(x) + N_X(x)\}.$$

Note that $\|h_0(x)\| = r(x)$ (see (2.1.2)). Since $h_0(x)$ is the orthogonal projection of the origin onto the set $\{\partial f(x) + N_X(x)\}$, it follows that

$$\langle h_0(x), h\rangle \geq \|h_0(x)\|^2 \ \ \forall h \in \partial f(x) + N_X(x). \tag{2.3.13}$$

Since $h_0(x) \in \partial f(x) + N_X(x)$, it follows that

$$\frac{1}{2} h_0(x) \in H(x). \tag{2.3.14}$$

Fix an arbitrary $x \notin X_s(\varepsilon(\cdot))$. By (2.1.3), we have

$$\|h_0(x)\| = r(x) > \varepsilon(x). \tag{2.3.15}$$

We further obtain

$$\begin{aligned}
\max_{h\in H(x)} \min_{g\in G_0(x)} \langle h, g\rangle &\geq \frac{1}{2} \min_{g\in G_0(x)} \langle h_0(x), g\rangle \\
&\geq \frac{1}{2} \min_{g\in\partial f(x)+N_X(x)+\varepsilon(x)B} \langle h_0(x), g\rangle \\
&\geq \frac{1}{2} \min_{\delta\in\varepsilon(x)B} \quad \min_{h\in\partial f(x)+N_X(x)} \langle h_0(x), h+\delta\rangle \\
&\geq \frac{1}{2} \min_{\delta\in\varepsilon(x)B} \langle h_0(x), h+\delta\rangle \\
&\geq \frac{1}{2} \min_{\delta\in\varepsilon(x)B} (\|h_0(x)\|^2 - \|\delta\|\|h_0(x)\|) \\
&\geq \frac{1}{2}\|h_0(x)\|(\|h_0(x)\| - \varepsilon(x)) > 0
\end{aligned}$$

where the first inequality follows from (2.3.14), the second inequality follows from (2.3.11), the fifth inequality follows from (2.3.13), and the last inequality follows from (2.3.15). Hence $x \notin \mathcal{A}_0$, and it follows that $\mathcal{A}_0 \subset X_s(\varepsilon(\cdot))$. Now applying Theorem 2.2.1 and Corollary 2.2.1, we immediately obtain the desired results. ∎

Adding the "heavy ball" term [51] in Algorithm 2.3.1, we arrive at the following modification of the parallel GGPM. In neural network literature, methods of this type are usually referred to as backpropagation with momentum term [24, 41].

Algorithm 2.3.2 (Parallel GGPM with heavy ball term). *Start with any* $x^0 \in X$. *Having* x^i, *compute* x^{i+1} *as follows :*

(•) Parallelization : *for each processor* $l \in \{1, \ldots, p\}$ *do*

$$z_l^{i,j+1} = z_l^{i,j} - \eta_i \tilde{g}_l^{i,j}, \quad j = 1, \ldots, K_l,$$

where $z_l^{i,1} = x^i$*.*

(•) Synchronization with heavy ball term :

$$x^{i+1} = \left[x^i + \sum_{l=1}^{p} (z_l^{i,K_l+1} - x^i) + \beta_i (x^i - x^{i-1}) \right]^+ .$$

With respect to coefficients multiplying the heavy ball term, we assume that

$$\beta_i \geq 0, \ i = 0, 1, \ldots, \ \ \beta_i \to 0. \tag{2.3.16}$$

We also make the following mild assumption on the stepsizes (in addition to (2.3.1), (2.3.5)) :

$$\limsup_{i \to \infty} \frac{\eta_{i-1}}{\eta_i} < +\infty. \tag{2.3.17}$$

The next result shows that methods with heavy ball term possess the same convergence and stability properties as the gradient projection methods.

Theorem 2.3.2 *For every sequence* $\{x^i\}$ *generated by Algorithm 2.3.2, all the conclusions of Theorem 2.3.1 hold.*

Proof. We show that the upper topological limits of the two characteristic mappings for Algorithms 2.3.1 and 2.3.2 are essentially the same (note that the mappings themselves are certainly different). We first define the following quantity

$$\mu_i := 2\beta_i KM\frac{\eta_{i-1}}{\eta_i}, \quad i = 1, 2, \dots, \quad \mu_0 = 0.$$

Note that by (2.3.16) and (2.3.17),

$$\mu_i \geq 0, \; i = 0, 1, \dots, \quad \lim_{i\to\infty} \mu_i = 0.$$

By the construction of the algorithm and (2.3.2),

$$\beta_i(x^i - x^{i-1}) \in x^i + 2\beta_i KM\eta_{i-1}B = x^i + \eta_i\mu_i B.$$

Let us denote the characteristic map of Algorithm 2.3.2 by $\tilde{G}(\cdot,\cdot)$. Then we have that

$$\tilde{G}(i, x^i) \subset G(i, x^i) + \mu_i B,$$

where $G(\cdot,\cdot)$ is the characteristic map of Algorithm 2.3.1 defined by (2.3.10). Every sequence $\{x^i\}$ generated by Algorithm 2.3.2 satisfies

$$x^{i+1} \in x^i - \eta_i\tilde{G}(i, x^i), \quad i = 0, 1, \dots, \quad x^0 \in X,$$

and hence also

$$x^{i+1} \in x^i - \eta_i\left(G(i, x^i) + \mu_i B\right).$$

Now taking into account that $\mu_i \to 0$, we obtain

$$\tilde{G}_0(x) := \overline{\mathrm{lt}}_{\substack{x'\to x\\ i\to\infty}} \tilde{G}(i, x^i) \subset \overline{\mathrm{lt}}_{\substack{x'\to x\\ i\to\infty}} (G(i, x') + \mu_i B) = \overline{\mathrm{lt}}_{\substack{x'\to x\\ i\to\infty}} G(i, x') = G_0(x).$$

Hence, by (2.3.11),

$$\tilde{G}_0(x) \subset \partial f(x) + N_X(x) + \varepsilon(x)B.$$

The rest of the proof is analogous to that of Theorem 2.3.1, and is thus omitted. ■

2.4 Important Special Cases

In this section we consider the standard optimization problem (2.1.1) of minimizing a Lipschitz continuous regular function over a convex compact set, and establish stronger convergence properties of GGPM in a number of important special cases. These include problems with relatively small perturbations, convex and strongly convex problems, and problems with weak sharp minima [51, 5].

We start with the following lemma which deals with the case of perturbations small relative to the residual function $r(\cdot)$ defined in (2.1.2).

Lemma 2.4.1 *Let*

$$\varepsilon(x) \leq \max\{\bar{\varepsilon}, \lambda r(x)\} \quad \forall x \in X,$$

where $\bar{\varepsilon} \geq 0,\ 1 > \lambda \geq 0$. *Then*

$$X_s(\varepsilon(\cdot)) \subset X_s(\bar{\varepsilon}).$$

In particular, if $\bar{\varepsilon} = 0$, *then*

$$X_s(\varepsilon(\cdot)) = X_s.$$

Proof. Suppose $x \in X_s(\varepsilon(x))$. Then, by (2.1.3) and the assumption of the lemma,

$$r(x) \leq \varepsilon(x) \leq \max\{\bar{\varepsilon}, \lambda r(x)\}.$$

If $\lambda r(x) \geq \bar{\varepsilon}$, then $r(x) \leq \lambda r(x)$ and $1 > \lambda \geq 0$ imply that $r(x) = 0$. Since $X_s(0) \subset X_s(\bar{\varepsilon})$, we have that $x \in X_s(\bar{\varepsilon})$. If $\lambda r(x) \leq \bar{\varepsilon}$, then $r(x) \leq \varepsilon(x) \leq \bar{\varepsilon}$, and hence $x \in X_s(\bar{\varepsilon})$. ■

Let $d(\cdot, C)$ be the distance function to the set $C \subset \Re^n$, that is

$$d(x, C) = \inf_{y \in C} \|x - y\|.$$

Define $\bar{\varepsilon} = \sup_{x \in X} \varepsilon(x)$, and $D = \sup_{x,y \in X} \|x - y\|$. The following lemma relates the ε-stationary sets to the ϵ-optimal sets for the case when $f(\cdot)$ is convex.

Lemma 2.4.2 *Let $f(\cdot)$ be convex on X. Then*

$$X_s(\varepsilon(x)) \subset X_{opt}\left(\varepsilon(x) d(x, X_{opt})\right).$$

In particular,

$$X_s(\varepsilon(\cdot)) \subset X_{opt}(\bar{\varepsilon} D).$$

If, in addition, $f(\cdot)$ is differentiable and strongly convex on X with modulus $\theta > 0$, and $X_s(\bar{\varepsilon}) \subset \operatorname{int} X$, then

$$X_s(\bar{\varepsilon}) \subset X_{opt}(\bar{\varepsilon}^2 / 2\theta).$$

Proof. Let $x \in X_s(\varepsilon(x))$. By definition of $X_s(\varepsilon(\cdot))$, there exist $g \in \partial f(x)$, $h_1 \in$

$N_X(x)$, and $h_2 \in \varepsilon(x)B$ such that $0 = g + h_1 + h_2$. Let x^* be the closest point to x in X_{opt}. By convexity of $f(\cdot)$, it follows that

$$\begin{aligned} f(x) - f(x^*) &\leq \langle -g, x^* - x\rangle \\ &= \langle h_1 + h_2, x^* - x\rangle \\ &\leq \langle h_2, x^* - x\rangle \\ &\leq \|h_2\|\|x^* - x\| \\ &\leq \varepsilon(x) d(x, X_{opt}), \end{aligned}$$

where the second inequality follows from definition of the normal cone. This establishes the first two assertions of the lemma.

For the last assertion, just note that ([51], p.24) for any $x \in X$

$$2\theta(f(x) - \min_{y \in X} f(y)) \leq \|\partial f(x)\|^2.$$

∎

Definition 2.4.1 *[5] We say that X_{opt} is a set of weak sharp minima with parameter $\rho > 0$ if*

$$f(x) - \min_{y \in X} f(y) \geq \rho d(x, X_{opt}) \;\; \forall x \in X.$$

The following important corollary shows that, for problems with weak sharp minima, certain ε-stationary sets coincide with the set of minima, provided ε is small relative to the parameter ρ given in Definition 2.4.1.

Corollary 2.4.1 *Let $f(\cdot)$ be convex on X. Assume that X_{opt} is a set of weak sharp minima with parameter $\rho > 0$. Then if*

$$\varepsilon(x) \le \max\{\nu, \lambda r(x)\} \ \ \forall x \in X, \ 0 \le \lambda < 1, \nu < \rho,$$

it follows that

$$X_s(\varepsilon(\cdot)) = X_{opt}.$$

Proof. Obviously, $X_{opt} \subset X_s(\varepsilon(\cdot))$. Take any $x \in X_s(\varepsilon(\cdot))$. By Lemmas 2.4.1 and 2.4.2, and our assumption, we have

$$x \in X_s(\varepsilon(\cdot)) \subset X_s(\nu) \subset X_{opt}(\nu d(x, X_{opt})).$$

Hence

$$\nu d(x, X_{opt}) \ge f(x) - \min_{y \in X} f(y) \ge \rho d(x, X_{opt}),$$

where the last inequality follows from Definition 2.4.1. Now $\nu < \rho$ implies that $d(x, X_{opt}) = 0$, hence $x \in X_{opt}$. ∎

When in Algorithms 2.3.1, 2.3.2 the parameter $K = 1$, those algorithms reduce to the following classical GGPM with the "heavy ball" term :

Algorithm 2.4.1 (GGPM with heavy ball term). *Start with any $x^0 \in X$. Having x^i, compute x^{i+1} as follows :*

$$x^{i+1} = \left[x^i - \eta_i(g_i + \delta(x^i, \alpha_i, i)) + \beta_i(x^i - x^{i-1})\right]^+$$

$$g_i \in \partial f(x^i, \alpha_i),\ i = 0, 1, \dots ,$$

where parameters $\{\eta_i\}$, $\{\alpha_i\}$, $\{\beta_i\}$ are the same as in Algorithms 2.3.1, 2.3.2.

From Theorems 2.3.1, 2.3.2, and Lemmas 2.4.1, 2.4.2, we immediately get the following results.

Theorem 2.4.1 *Every sequence $\{x^i\}$ generated by Algorithm 2.4.1 possesses the following properties :*

1. *there exists an $f(\cdot)$-connected component $X_s(\varepsilon(\cdot))^\gamma$ of $X_s(\varepsilon(\cdot))$ such that*

$$\bar{\mathrm{lt}}_{i\to\infty}\{f(x^i)\} = f\left(\bar{\mathrm{lt}}\{x^i\} \cap X_s(\varepsilon(\cdot))^\gamma\right);$$

2. *every subsequence $\{x^{i_m}\}$ of $\{x^i\}$ satisfying (2.3.12) converges into $X_s(\varepsilon(\cdot))^\gamma$;*

3. *if perturbations are relatively small, that is*

$$\varepsilon(x) \le \lambda r(x)\ \ \forall x \in X,\ \ 0 \le \lambda < 1,$$

and the set $f(X_s)$ is nowhere dense in $\Re$, then $\{x^i\}$ converges into X_s;

4. *if $f(\cdot)$ is convex, then $\{x^i\}$ converges into the set*

$$X_{opt}\left(\varepsilon(x) d(x, X_{opt})\right) \subset X_{opt}(\bar{\varepsilon} D);$$

5. *if $f(\cdot)$ is convex, X_{opt} is a set of weak sharp minima with parameter $\rho > 0$, and*

$$\varepsilon(x) < \rho\ \ \forall x \in X,$$

then $\{x^i\}$ converges into X_{opt}.

6. if $f(\cdot)$ is strongly convex with modulus $\theta > 0$, and $X_s(\bar{\varepsilon}) \subset \text{int } X$ ($\bar{\varepsilon} := \sup_{x \in X} \varepsilon(x)$), then $\{x^i\}$ converges into $X_{opt}(\bar{\varepsilon}^2/2\theta)$.

Remark 2.4.1 *Theorem 2.4.1 extends and strengthens the results on convergence properties of the generalized gradient projection method given in [46, 11, 75].*

2.5 Applications to Neural Network Training

In this section we briefly describe how results of Section 2.3 can be applied to reveal some important properties of various neural network learning techniques. In particular, we give a precise characterization to empirically observed stability of neural networks and backpropagation training [58, 19]. We also show that the properties of the weight perturbation [23] algorithm can be derived by making use of the presented analysis.

2.5.1 Backpropagation With Noise

We note that when implemented in hardware, BP algorithm is likely to have some kind of electronic imperfections [23]. Faults in multiplier circuits introduce errors when function and gradient values are propagated through the network. Therefore, in practical electronic implementations, even when the input data can be considered to be free of noise, the algorithm is influenced by certain perturbations induced by hardware limitations. It is therefore desirable to use algorithms that are tolerant to imperfections in a neural network chip. This makes error-stability analysis of training techniques of practical importance.

We regard training artificial neural network as optimization of a certain error function (see Section 1.1). Note that $f(x, \alpha)$ given by (1.1.2)-(1.1.3) is precisely of the form (2.1.5) that was studied in this chapter.

Each iteration of the **serial online** BP consists of a step in the direction of negative gradient $-\nabla f_j$ of a partial error function f_j associated with the j-th training example. Thus BP is a special case of Algorithm 2.3.1. Many other computationally important BP modifications, such as **parallel** BP [42, 49, 14], BP with **momentum term** [24], and BP with varying smoothing parameter [65] all fall within the framework of Section 2.3.

It is quite common that for a sample ξ^j in the training set some of its attributes (i.e. the components of the m-dimensional vector) are computed (or supplied) with an error that we shall denote Δ_j. Obviously, this induces certain perturbation in values of the corresponding error function f_j and its gradient. We can then write (see (1.1.3))

$$\tilde{f}_j(w, \theta, v, \tau, \alpha) := \left(s\left(\sum_{i=1}^{h} s((\xi^j + \Delta_j)w^i - \theta^i, \alpha)v^i - \tau, \alpha \right) - t^j \right)^2,$$

and

$$\nabla \tilde{f}_j(x, \alpha) = \nabla f_j(x, \alpha) + \delta_j(x, \alpha).$$

Note that it is fairly straightforward to estimate the dependence of δ_j on Δ_j.

We can then introduce the exact asymptotic level of perturbations (2.3.7) by

$$\varepsilon(x) = \limsup_{\substack{z_j(\in X)\to x \\ i\to\infty}} \| \sum_{j\in Q} \delta_j(z_j, \alpha_i, i)\|,$$

where Q is the set of training examples with noise. If some upper bound on $\Delta_j,\ j \in Q$ is known then the corresponding perturbations $\delta_j,\ j \in Q$ and their asymptotic level $\varepsilon(\cdot)$ can be estimated. This in turn yields the guaranteed $\varepsilon(\cdot)$-stationarity of all the accumulation points of the BP iterates.

One useful technique that sometimes improves the performance of the neural network is deliberate adding of some noise to the input training set. It appears that a neural network trained with some induced noise often has a better ability to recognize noisy patterns, and performs better in classifying patterns that were not presented to the network during the training procedure [60]. The last property is usually called the *generalization* ability and it is one of the major strengths of artificial neural networks.

As another source of perturbations in the neural network training, we note the technique presented in [19]. To simplify the network topology and improve the network generalization properties, it is proposed in [19] to eliminate at the late stages of training the arcs with sufficiently small weights. The latter is equivalent to forcing the corresponding weights to zero, and can also be treated as induced perturbations.

We finally mention the *node perturbation* approach first proposed in [71]. Although detailed technical analysis of this algorithm is beyond the scope of this work, we note that the framework presented in this chapter provides a useful tool for such analysis.

2.5.2 The Weight Perturbation Algorithm

It should be noted that some algorithms work properly in computer simulations for "ideal" (or theoretical) neural networks, but their performance becomes unsatisfactory in practical hardware implementations. One of the algorithms that is less sensitive to the hardware limitations due to the chip and network interface, is the so-called weight perturbation (WP) training [23]. This technique is essentially an incremental gradient-type method where the gradient is approximated by one of the finite difference techniques. In its simplest form, WP is the following algorithm

$$x^{i+1} = x^i - \eta_i g_j^i, \quad i = 0, 1, \dots,$$

where

$$g_j^i = \left(\frac{f_j(x^i + \Delta_1^i e_1) - f_j(x^i)}{\Delta_1^i}, \dots, \frac{f_j(x^i + \Delta_n^i e_n) - f_j(x^i)}{\Delta_n^i} \right), \quad j = i \bmod K,$$

and $\{e_1, \dots, e_n\}$ is the standard basis of $\Re^n$. Clearly, WP needs more pattern presentations than BP. However, in [23] it is reported that sometimes implementation of BP requires excessive computational hardware, and WP is more economical for parallel analog implementations. Thus speed is traded for more reliable and practical hardware.

We point out that there has been no rigorous analysis of WP training in the literature. We can readily apply results presented in this chapter to derive the properties of WP. In particular, for the forward difference scheme above, it is well known that

$$\|\delta_j(x^i)\| = O\left(\max_{k=1,\dots,n} \Delta_k^i\right),$$

where

$$\delta_j(x^i) = \nabla f_j(x^i) - g_j^i.$$

Hence

$$\varepsilon(x) = \limsup_{\substack{z_j(\in X)\to x \\ i\to\infty}} \|\sum_{j=1}^{K} \delta_j(z_j)\| \le c \ \limsup_{i\to\infty} \max_{k=1,\dots,n} \Delta_k^i$$

for some constant $c > 0$.

The above estimate can be further improved if the central difference approximation is used. In that case,

$$\|\delta(x^i)\| = O\left((\max_{k=1,\dots,n} \Delta_k^i)^2\right).$$

The central difference scheme yields further increase in the number of training samples presentations (the number of the partial error functions evaluations). However, for some problems this is still practical [23].

Chapter 3

Convergence Analysis of Perturbed Feasible Descent Methods

We develop a general approach to convergence analysis of feasible descent methods in the presence of perturbations [63]. The important novel feature of our analysis is that perturbations need not tend to zero in the limit. In that case, standard convergence analysis techniques are not applicable. Therefore a new approach is needed. We show that, in the presence of perturbations, a certain ε-approximate solution can be obtained, where ε depends on the level of perturbations linearly. Applications to the gradient projection, proximal minimization and extragradient algorithms are described.

3.1 Introduction

We consider the general mathematical programming problem of minimizing a differentiable function $f : \Re^n \to \Re$ over a closed convex set X in $\Re^n$:

$$\min_{x \in X} f(x). \tag{3.1.1}$$

We assume that $f \in C^1_L(X)$, that is $f(\cdot)$ has Lipschitz continuous partial derivatives :

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x \in X,\ y \in X, \tag{3.1.2}$$

where L is a positive scalar, $\nabla f(\cdot)$ denotes the gradient of $f(\cdot)$, and $\|\cdot\|$ denotes the Euclidean norm.

Let $[\cdot]^+$ denote the orthogonal projection onto X. Following [31], we consider a broad class of feasible descent methods that can be represented by the formula

$$x^{new} := [x - \eta\nabla f(x) + e(x)]^+, \tag{3.1.3}$$

where η is a positive scalar, and mapping $e : \Re^n \to \Re^n$ is the defining feature of each particular algorithm (see Section 3.3). This is a rather general framework that includes a gradient projection algorithm [15, 27]; proximal minimization algorithm [44, 55]; and the extragradient algorithm [25, 43] among others. We note, in the passing, that for characteristic mappings $e(\cdot)$ of feasible descent methods, $e(x^i) \to 0$ as $i \to \infty$ by algorithm construction [31].

In this chapter, we are concerned with the behaviour of feasible descent algorithms in the presence of perturbations :

$$x^{new} := [x - \eta \nabla f(x) + e(x) + \delta(x)]^{+}. \tag{3.1.4}$$

Here $e(\cdot)$ plays the same role as in (3.1.3), namely it is the characteristic of the method, while $\delta(\cdot)$ represents perturbations due to inexact computation of the gradient of $f(\cdot)$, or inexact subproblem solution, or both. We say that perturbations are **essential** (nonvanishing) if

$$\delta(x^i) \not\to 0 \text{ as } i \to \infty.$$

In this chapter, we consider nonvanishing perturbations and make only a mild assumption that perturbations are uniformly bounded :

$$\|\delta(x)\| \leq \bar{\varepsilon} \quad \text{for some } \bar{\varepsilon} > 0, \quad \forall x \in X. \tag{3.1.5}$$

The latter is the only practical assumption in the case when perturbations cannot be effectively controlled. This may happen, for example, when the function and/or gradient values are not given explicitly, but instead are computed as an approximate solution of some, possibly difficult, subproblem. We note that very little is known about convergence properties of essentially perturbed algorithms. The primary contribution of this chapter is laying down theoretical framework for analysis of such algorithms.

Convergence (and rate of convergence) of feasible descent methods have been studied extensively (see [31] and references therein). We point out that the previous work either deals with the case when no perturbations are present

$(\delta(x^i) = 0)$, or assumes some conditions that explicitly or implicitly imply that perturbations vanish in the limit $(\delta(x^i) \to 0)$. Some conditions of this type have been used in the analysis of matrix splitting methods [35, 30] :

$$\|\delta(x^i)\| \leq c\|x^{i+1} - x^i\|, \quad c > 0,\ c \text{ sufficiently small}$$

or

$$\sum_{i=0}^{\infty} \delta(x^i) < \infty.$$

Note that under either assumption, $\delta(x^i) \to 0$ as $i \to \infty$. In these cases, convergence properties of the algorithm stay intact, except possibly for the rate of convergence. We emphasize that the setting considered in this work is fundamentally different. Condition (3.1.5) no longer guarantees convergence of iterates generated by (3.1.4) to an exact solution of (3.1.1). Moreover, standard relations such as

$$f(x^i) - f(x^{i+1}) \geq 0,$$

and

$$\|x^{i+1} - x^i\| \to 0 \text{ as } i \to \infty$$

need not hold (see Section 3.2). This makes traditional convergence analysis techniques [51, 50] inapplicable. In this chapter, we develop a new approach to the analysis of algorithms with nonvanishing perturbations.

Our analysis extends some of the ideas presented in [4, 42, 74] for methods of unconstrained optimization. Essential perturbations were considered in [62] in a different context of incremental gradient-type methods with decaying stepsize.

A special case of an approximate gradient projection method with decaying stepsize is also studied in [32]. We note that in this chapter, the stepsize η is bounded away from zero. Therefore, the situation and the analysis required are completely different from [62, 32].

We now define the following residual function

$$r(x) := x - [x - \nabla f(x)]^+.$$

It is well known that some $\bar{x} \in \Re^n$ satisfies the Minimum Principle optimality condition [34] for problem (3.1.1) if and only if $r(\bar{x}) = 0$. We shall call such $\bar{x}$ a stationary point of (3.1.1). For a nonnegative upper semicontinuous function $\varepsilon : \Re^n \to \Re_+$, we define an **$\varepsilon(\cdot)$-stationary** set of problem (3.1.1) as follows :

$$X_s(\varepsilon(\cdot)) := \{x \in X \mid \quad \|r(x)\| \leq \varepsilon(x)\}. \tag{3.1.6}$$

Clearly, $X_s(0)$ is the set of all stationary points in the usual sense (we shall use the notation $X_s := X_s(0)$). In Section 3.2, we show that for any sequence of iterates generated by (3.1.4), there exists at least one accumulation point which is in the set $X_s(\epsilon)$ with ϵ depending on the level of perturbations linearly.

We note that another important property of the residual function $r(x)$ is that, under certain conditions, its norm provides a (local) upper bound on the distance to the set X_s [31, 53]. Namely, there exist positive constants μ and ν (depending on $f(\cdot)$ and X only) such that

$$d(x, X_s) \leq \mu \|r(x)\| \quad \forall x \text{ with } \|r(x)\| \leq \nu, \tag{3.1.7}$$

where $d(\cdot, X_s)$ denotes the Euclidean distance to X_s. Moreover, under additional assumptions, this condition holds with $\nu = \infty$ (global error bound) [29, 48, 28]. Therefore, if $x \in X_s(\epsilon)$ and the bound (3.1.7) holds with $\nu \geq \epsilon$, it follows immediately that

$$d(x, X_s) \leq \mu \|r(x)\| \leq \mu\epsilon.$$

The rest of the chapter is organized as follows. In Section 3.2 we develop our general technique for convergence analysis of perturbed algorithms. In Section 3.3 we show how our results apply to the gradient projection, proximal point and extragradient algorithms. Section 3.4 contains some concluding remarks.

One more word about our notation. For a bounded sequence $\{x^i\}$ in $\Re^n$, $\bar{\mathrm{lt}}_{i\to\infty}\{x^i\}$ denotes the set of all accumulation points of $\{x^i\}$.

3.2 Convergence Analysis of Methods With Perturbations

In this Section, we present our general framework for the analysis of feasible descent methods in the presence of essential perturbations. Our argument is based on monitoring the behaviour of $f(\cdot)$ on the iterates of the algorithm. We emphasize that this behaviour is nonmonotone, and Lyapunov-type convergence analysis [50, 72] cannot be applied.

We first state three well known results that will be used later.

Lemma 3.2.1 *([51],p.6) Let* $f(\cdot) \in C^1_L(X)$ *, then*

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in X.$$

Lemma 3.2.2 *([51],p.121) For any* $x \in \Re^n$*, any* $y \in \Re^n$*, and any* $z \in X$ *the following relations hold*

$$\langle y - [y]^+, z - [y]^+ \rangle \leq 0,$$

$$\|[x]^+ - [y]^+\| \leq \|x - y\|.$$

Lemma 3.2.3 *([13], Lemma 1) For any* $x \in \Re^n$*, any* $y \in \Re^n$*, and any* $\eta > 0$

$$\max\{1, \eta\}\|x - [x - y]^+\| \geq \|x - [x - \eta y]^+\| \geq \min\{1, \eta\}\|x - [x - y]^+\|.$$

The method under consideration is the following model algorithm.

Algorithm 3.2.1 *Start with any* $x^0 \in X$*. For* $i = 0, 1, 2, \ldots$ *let*

$$x^{i+1} \in T(x^i),$$

where

$$T(x) = [x - \eta \nabla f(x) + e(x) + \delta(x)]^+,$$

and the following conditions are satisfied

$$\|e(x)\| \leq c_1 \|x - T(x)\|, \quad 0 \leq c_1 < 1, \tag{3.2.1}$$

$$\langle e(x), x - T(x) \rangle \geq -c_2 \|x - T(x)\|^2, \quad 0 \leq c_2 < 1, \tag{3.2.2}$$

$$c_3 \le \liminf_i \eta_i, \quad \limsup_i \eta_i \le \min\{1, \frac{2(1-c_2)}{L} - c_3\}, \tag{3.2.3}$$

where

$$0 < c_3 < \frac{1-c_2}{L}.$$

In Section 3.3, we show that various important optimization methods fall within the framework of Algorithm 3.2.1. Condition (3.2.1) is standard for feasible descent methods and is a consequence of algorithm construction [31]. Bounds (3.2.3) imposed on the stepsize are also fairly standard. With respect to (3.2.2), we note the following. If the left-hand-side of (3.2.2) is nonnegative for all x then we set $c_2 = 0$, otherwise $c_2 = c_1$ (it follows that $0 \le c_2 < 1$).

To study the convergence properties of Algorithm 3.2.1, we need to estimate the level of perturbations in the limit. We say that $\varepsilon(x)$ is the *exact asymptotic level of perturbations* at a point $x \in X$, if

$$\varepsilon(x) = \limsup_{\substack{y^k (\in X) \to x \\ k \to \infty}} \|\delta(y^k)\|.$$

It is easy to see that $\varepsilon(\cdot) : \Re^n \to \Re_+$ is upper semicontinuous.

For the clarity of presentation, we briefly outline our argument. Using Lemmas 3.2.1-3.2.3 and conditions (3.2.1)-(3.2.3), we show that $f(x) - f(T(x)) \ge \varphi(x)$, where $\varphi(x)$ is a certain lower semicontinuous function which depends on the residual $r(x)$ and the asymptotic level of perturbations $\varepsilon(x)$ (note that $\varphi(\cdot)$ need not be nonnegative). If $f(\cdot)$ is bounded from below on X, then for any sequence of iterates generated by Algorithm 3.2.1 there must exist at least one accumulation point belonging to the level set $\{x \in X \mid \varphi(x) \le 0\}$ (otherwise we get a contradiction). Finally, using the dependence of $\varphi(\cdot)$ on $r(\cdot)$ and $\varepsilon(\cdot)$, we establish a certain relationship between the level sets of $\varphi(\cdot)$ and the $\varepsilon(\cdot)$-stationary sets (3.1.6) of problem (3.1.1).

We are now ready to state and prove our main result.

Theorem 3.2.1 *Suppose $f \in C^1_L(X)$ and $f(\cdot)$ is bounded from below on X. Let conditions (3.2.1)-(3.2.3) be satisfied. Then there exist positive constants d_1 and d_2 such that :*

For every bounded sequence $\{x^i\}$ generated by Algorithm 3.2.1, there exists an accumulation point $\bar{x}$ of $\{x^i\}$ such that

$$\bar{x} \in X_s\left(d_1\varepsilon(\cdot)\right).$$

For every subsequence $\{x^{i_m}\}$ of $\{x^i\}$ satisfying

$$\limsup_{m\to\infty} f(x^{i_m}) \leq \liminf_{i\to\infty} f(x^i) + t \quad \textit{for some } t \geq 0,$$

it follows that

$$\operatorname*{\bar{lt}}_{m\to\infty}\{x^{i_m}\} \subset X_s\left(d_1\varepsilon(\cdot) + d_2 t^{\frac{1}{2}}\right).$$

In particular, if the sequence $\{f(x^i)\}$ converges, then

$$\operatorname*{\bar{lt}}_{i\to\infty}\{x^i\} \subset X_s\left(d_1\varepsilon(\cdot)\right).$$

Proof. Let $x := x^i$. Then for every $i = 0, 1, 2, \ldots$, by Lemma 3.2.1,

$$f(x) - f(T(x)) \geq -\langle \nabla f(x), T(x) - x\rangle - \frac{L}{2}\|T(x) - x\|^2. \tag{3.2.4}$$

By Lemma 3.2.2 (taking $y = x - \eta_i \nabla f(x) + e(x) + \delta(x)$ and $z = x \in X$), we have

$$\langle x - \eta_i \nabla f(x) + e(x) + \delta(x) - T(x), x - T(x) \rangle \leq 0.$$

Hence

$$-\langle \nabla f(x), T(x) - x \rangle \geq \frac{1}{\eta_i} \left(\|x - T(x)\|^2 + \langle e(x) + \delta(x), x - T(x) \rangle \right).$$

Using (3.2.2), we have

$$-\langle \nabla f(x), T(x) - x \rangle \geq \frac{1}{\eta_i} \left((1 - c_2)\|x - T(x)\|^2 + \langle \delta(x), x - T(x) \rangle \right).$$

Combining the latter inequality with (3.2.4), we further obtain

$$\begin{aligned} f(x) - f(T(x)) &\geq \left(\frac{1 - c_2}{\eta_i} - \frac{L}{2} \right) \|T(x) - x\|^2 + \frac{1}{\eta_i} \langle \delta(x), x - T(x) \rangle \\ &\geq \left(\frac{1 - c_2}{\eta_i} - \frac{L}{2} \right) \|T(x) - x\|^2 - \frac{1}{\eta_i} \|\delta(x)\| \|x - T(x)\| \\ &\geq \frac{1}{\eta_i}(1 - c_2 - \frac{L\eta_i}{2}) \|T(x) - x\|^2 - \frac{1}{\eta_i} \varepsilon(x) \|T(x) - x\| \\ &\geq \frac{L^2 c_3}{4(1 - c_2) - 2Lc_3} \|T(x) - x\|^2 \\ &\quad - \frac{1}{c_3} \varepsilon(x) \|T(x) - x\|, \end{aligned} \tag{3.2.5}$$

where the second relation follows from the Cauchy-Schwartz inequality, the third inequality follows from the definition of $\varepsilon(\cdot)$, and the last inequality follows from (3.2.3) for i sufficiently large, say $i \geq i_1$.

By Lemmas 3.2.3 and 3.2.2, the triangle inequality, and (3.2.1), it follows that

$$\begin{aligned}
\min\{1,\eta_i\}\|r(x)\| &\leq \|x-[x-\eta_i\nabla f(x)]^+\| \\
&\leq \|x-T(x)\|+\|T(x)-[x-\eta_i\nabla f(x)]^+\| \\
&\leq \|x-T(x)\|+\|e(x)+\delta(x)\| \\
&\leq (1+c_1)\|x-T(x)\|+\varepsilon(x).
\end{aligned}$$

For $i \geq i_1$, using (3.2.3), we obtain

$$\|x-T(x)\| \geq \frac{1}{1+c_1}\left(c_3\|r(x)\|-\varepsilon(x)\right). \tag{3.2.6}$$

Similarly,

$$\begin{aligned}
\|x-T(x)\| &= \|x-[x-\eta_i\nabla f(x)]^+ + [x-\eta_i\nabla f(x)]^+ - T(x)\| \\
&\leq \max\{1,\eta_i\}\|r(x)\|+\|e(x)+\delta(x)\| \\
&\leq \|r(x)\|+c_1\|x-T(x)\|+\varepsilon(x).
\end{aligned}$$

Hence

$$\|x-T(x)\| \leq \frac{1}{1-c_1}\left(\|r(x)\|+\varepsilon(x)\right). \tag{3.2.7}$$

For $i \geq i_1$, combining (3.2.5)-(3.2.7) yields

$$\begin{aligned}
f(x)-f(T(x)) &\geq \frac{L^2c_3}{2(1+c_1)^2(2(1-c_2)-Lc_3)}\left(c_3\|r(x)\|-\varepsilon(x)\right)^2 \\
&\quad -\frac{1}{c_3(1-c_1)}\varepsilon(x)\left(\|r(x)\|+\varepsilon(x)\right) \\
&= b_1\|r(x)\|^2-b_2\varepsilon(x)\|r(x)\|-b_3\varepsilon(x)^2,
\end{aligned}$$

where

$$b_1 := \frac{L^2 c_3^3}{2(1+c_1)^2(2(1-c_2)-Lc_3)},$$

$$b_2 := \frac{L^2 c_3^2}{(1+c_1)^2(2(1-c_2)-Lc_3)} + \frac{1}{c_3(1-c_1)},$$

$$b_3 := \frac{1}{c_3(1-c_1)} - \frac{L^2 c_3}{2(1+c_1)^2(2(1-c_2)-Lc_3)}.$$

By (3.2.1)-(3.2.3), it is easy to see that $b_1 > 0$ and $b_2 > 0$. We next check that $b_3 > 0$. By (3.2.3),

$$\frac{1}{c_3} \geq \frac{L}{2(1-c_2)-Lc_3}.$$

Hence

$$\frac{L^2 c_3}{2(1+c_1)^2(2(1-c_2)-Lc_3)} \leq \frac{L}{2} < \frac{1-c_2}{c_3} < \frac{1}{c_3(1-c_1)},$$

where the second inequality follows from (3.2.3). Hence $b_3 > 0$.

We next define the following auxiliary function $\varphi : X \to \Re$ which is crucial for our analysis

$$\varphi(x) := b_1\|r(x)\|^2 - b_2\varepsilon(x)\|r(x)\| - b_3\varepsilon(x)^2.$$

With this definition, we have

$$f(x) - f(T(x)) \geq \varphi(x). \tag{3.2.8}$$

It is easy to see that since $\|r(\cdot)\|$ is continuous, $b_2 > 0$, $b_3 > 0$, and $\varepsilon(\cdot)$ is nonnegative and upper semicontinuous, then $\varphi(\cdot)$ is lower semicontinuous. We shall consider the level sets of $\varphi(\cdot)$ defined as

$$\mathcal{L}(\varphi, t) := \{x \in X \mid \varphi(x) \leq t\}, \quad t \geq 0. \tag{3.2.9}$$

Note that the set $\mathcal{L}(\varphi, t)$ is closed for any $t \in \Re$ (Theorem 7.1, [54]). Denoting $u = \|r(x)\|$, $\varepsilon = \varepsilon(x)$, and resolving the quadratic inequality in u

$$b_1 u^2 - b_2 \varepsilon u - b_3 \varepsilon^2 - t \leq 0,$$

we conclude that

$$u \leq \frac{b_2 \varepsilon}{2b_1} + \frac{1}{2b_1}\sqrt{(b_2^2 + 4b_1 b_3)\varepsilon^2 + 4b_1 t}.$$

Hence

$$\mathcal{L}(\varphi, t) = \{x \in X \mid \|r(x)\| \leq \frac{b_2 \varepsilon(\cdot)}{2b_1} + \frac{1}{2b_1}\sqrt{(b_2^2 + 4b_1 b_3)\varepsilon(\cdot)^2 + 4b_1 t}\}.$$

In particular,

$$\mathcal{L}(\varphi, 0) = \{x \in X \mid \|r(x)\| \leq \frac{b_2 + \sqrt{b_2^2 + 4b_1 b_3}}{2b_1}\varepsilon(\cdot)\}.$$

Defining

$$d_1 := \frac{b_2 + \sqrt{b_2^2 + 4b_1 b_3}}{2b_1},$$

and

$$d_2 := b_1^{-\frac{1}{2}},$$

and taking into account the definition of $X_s(\varepsilon(\cdot))$, we further conclude that

$$\mathcal{L}(\varphi, t) \subset X_s\left(d_1 \varepsilon(\cdot) + d_2 t^{\frac{1}{2}}\right), \tag{3.2.10}$$

and

$$\mathcal{L}(\varphi, 0) = X_s\left(d_1 \varepsilon(\cdot)\right). \tag{3.2.11}$$

We next prove that there exists an accumulation point $\bar{x}$ of $\{x^i\}$ such that $\bar{x} \in \mathcal{L}(\varphi, 0)$. Suppose the opposite holds. By (3.2.8), we have

$$f(x^i) - f(x^{i+1}) \geq \varphi(x^i), \quad \forall i \geq i_1.$$

Since by our assumption, $\bar{\mathrm{lt}}_{i\to\infty}\{x^i\} \cap \mathcal{L}(\varphi, 0) = \emptyset$, it follows from (3.2.9) and lower semicontinuity of $\varphi(\cdot)$ that for some i_2 sufficiently large, and some $c > 0$,

$$\varphi(x^i) \geq c > 0 \quad \forall i \geq i_2.$$

Denoting $k := \max\{i_1, i_2\}$, for any $i > k$, we have

$$f(x^k) - f(x^i) = \sum_{j=k}^{i-1}(f(x^j) - f(x^{j+1})) \geq \sum_{j=k}^{i-1} c = (i-k)c.$$

Letting $i \to \infty$, we get that $\{f(x^i)\} \to -\infty$ which contradicts the fact that $f(\cdot)$ is bounded from below on X. Hence the assumption is invalid, and $\bar{\mathrm{lt}}_{i\to\infty}\{x^i\} \cap \mathcal{L}(\varphi, 0) \neq \emptyset$. Now the first assertion of the theorem follows from (3.2.11).

Consider now a subsequence $\{x^{i_m}\}$ of $\{x^i\}$, and a $t \geq 0$ such that

$$\limsup_{m\to\infty} f(x^{i_m}) \leq \liminf_{i\to\infty} f(x^i) + t.$$

We shall establish that

$$\bar{\mathrm{lt}}_{m\to\infty}\{x^{i_m}\} \subset \mathcal{L}(\varphi, t).$$

Suppose this is not true. Then (passing onto a subsequence, if necessary) $\{x^{i_{m_k}}\} \to y \notin \mathcal{L}(\varphi, t)$. Therefore, by (3.2.9), for some $c > 0$,

$$\varphi(y) \geq t + 2c.$$

By lower semicontinuity of $\varphi(\cdot)$, there exists k_1 sufficiently large such that

$$\varphi(x^{i_{m_k}}) \geq t + c \quad \forall k \geq k_1.$$

Let $k_2 := \min\{k \mid i_{m_k} \geq i_1\}$. By (3.2.8),

$$f(x^{i_{m_k}}) - f(x^{i_{m_k}+1}) \geq t + c, \quad \forall k \geq \max\{k_1, k_2\}. \tag{3.2.12}$$

Also, since $\{x^{i_{m_k}}\} \to y$,

$$f(y) = \lim_{k\to\infty} f(x^{i_{m_k}}) \leq \limsup_{m\to\infty} f(x^{i_m}) \leq \liminf_{i\to\infty} f(x^i) + t. \tag{3.2.13}$$

Combining the last relation with (3.2.12), we have

$$\begin{aligned}
\liminf_{i\to\infty} f(x^i) &\leq \limsup_{k\to\infty} f(x^{i_{m_k}+1}) \\
&\leq \limsup_{k\to\infty} f(x^{i_{m_k}}) - t - c \\
&= \lim_{k\to\infty} f(x^{i_{m_k}}) - t - c \\
&= f(y) - t - c \\
&< f(y) - t,
\end{aligned}$$

which contradicts (3.2.13). Hence $\bar{\mathrm{l}}\mathrm{t}_{m\to\infty}\{x^{i_m}\} \subset \mathcal{L}(\varphi, t)$, and the second assertion of the theorem follows from (3.2.10).

For the last assertion note that if the sequence $\{f(x^i)\}$ converges, then for *every* subsequence $\{x^{i_m}\}$ of $\{x^i\}$ it follows that

$$\limsup_{m\to\infty} f(x^{i_m}) = \liminf_{i\to\infty} f(x^i).$$

Hence

$$\bar{\text{lt}}_{i\to\infty} \{x^i\} \subset \mathcal{L}(\varphi, 0),$$

and the last assertion of the theorem follows from (3.2.11).

The proof is complete. ∎

Remark 3.2.1 *If* $\limsup_i \varepsilon(x^i) \le \epsilon$, *and the error bound (3.1.7) holds with* $\nu \ge \epsilon$, *then it follows that there exist an accumulation point* $\bar{x}$ *of the sequence* $\{x^i\}$ *and a stationary point* $\hat{x} \in X_s$ *such that*

$$\|\bar{x} - \hat{x}\| \le \mu\epsilon,$$

where μ *is as specified in (3.1.7).*

3.3 Applications

In this Section, we briefly discuss applications of our analysis to a number of well known algorithms.

3.3.1 Gradient Projection Algorithm

We first consider the gradient projection algorithm [15, 27]. In the presence of perturbations, it takes the following form

$$x^{i+1} = [x^i - \eta_i \nabla f(x^i) + \delta(x^i)]^+.$$

Obviously, this method is a special case of Algorithm 3.2.1 corresponding to

$$e(x) = 0 \quad \forall x \in X.$$

Consequently, we can take $c_1 = 0$, and $c_2 = 0$ in (3.2.1)-(3.2.3). Provided the stepsize satisfies the standard conditions

$$0 < c_3 \leq \eta_i \leq \frac{2}{L} - c_3,$$

it can be verified that

$$d_1 = O(L^2),$$

where d_1 is the constant involved in Theorem 3.2.1.

3.3.2 Proximal Minimization Algorithm

Given a current iterate x^i, the proximal minimization algorithm [44, 55] generates the next iterate x^{i+1} according to

$$x^{i+1} = \arg\min_{x \in X} \psi_i(x) := f(x) + \frac{1}{2\eta_i}\|x - x^i\|^2.$$

This method also falls within the presented framework as can be seen from the following. If the subproblems above are solved exactly, then the gradient projection optimality condition is satisfied

$$x^{i+1} = [x^{i+1} - c\nabla\psi_i(x^{i+1})]^+ \quad \forall c > 0.$$

Suppose only approximate solutions to the subproblems are available. Then we have

$$
\begin{aligned}
x^{i+1} &= [x^{i+1} - \eta_i \nabla \psi_i(x^{i+1}) + \delta(x^i)]^+ \\
&= [x^{i+1} - \eta_i \left(\nabla f(x^{i+1}) + \frac{1}{\eta_i}(x^{i+1} - x^i) \right) + \delta(x^i)]^+ \\
&= [x^i - \eta_i \nabla f(x^{i+1}) + \delta(x^i)]^+ \\
&= [x^i - \eta_i \nabla f(x^i) + e(x^i) + \delta(x^i)]^+,
\end{aligned}
$$

where

$$
e(x^i) = \eta_i \left(\nabla f(x^i) - \nabla f(x^{i+1}) \right).
$$

Since, by the Lipschitz continuity of the gradient,

$$
\|e(x^i)\| \le \eta_i L \|x^i - x^{i+1}\|,
$$

it is easy to see that (3.2.1)-(3.2.3) are satisfied provided $\limsup_i \eta_i < 1/L$. If $f(\cdot)$ is convex then

$$
\langle e(x^i), x^i - x^{i+1} \rangle = \eta_i \langle \nabla f(x^i) - \nabla f(x^{i+1}), x^i - x^{i+1} \rangle \ge 0,
$$

and we can further take $c_2 = 0$. It can be checked that

$$
d_1 = O(L^2),
$$

where d_1 is the constant involved in Theorem 3.2.1.

3.3.3 Extragradient Method

Consider now the extragradient method [25, 43] which updates a current iterate according to the double-projection formula

$$x^{i+1} = [x^i - \eta_i \nabla f \left([x^i - \eta_i \nabla f(x^i)]^+ \right)]^+.$$

This iteration can be re-written as

$$x^{i+1} = [x^i - \eta_i \nabla f(x^i) + e(x^i)]^+,$$

where

$$e(x^i) = \eta_i \left(\nabla f(x^i) - \nabla f([x^i - \eta_i \nabla f(x^i)]^+) \right).$$

In the presence of perturbations, we have

$$x^{i+1} = [x^i - \eta_i \nabla f(x^i) + e(x^i) + \delta(x^i)]^+,$$

where $\delta(x^i)$ is the aggregate perturbation at the i-th iteration. Let $y^i = [x^i - \eta_i \nabla f(x^i)]^+$. By the Lipschitz continuity of the gradient, we have

$$\begin{aligned} \|e(x^i)\| &= \eta_i \|\nabla f(y^i) - \nabla f(x^i)\| \\ &\leq \eta_i L \|y^i - x^i\|. \end{aligned}$$

Furthermore,

$$\begin{aligned} \|x^{i+1} - x^i\| &\geq \|x^i - y^i\| - \|y^i - x^{i+1}\| \\ &= \|x^i - y^i\| - \|[x^i - \eta_i \nabla f(x^i)]^+ - [x^i - \eta_i \nabla f(y^i)]^+\| \\ &\geq \|x^i - y^i\| - \eta_i \|\nabla f(x^i) - \nabla f(y^i)\| \\ &\geq (1 - \eta_i L)\|x^i - y^i\|, \end{aligned}$$

where the second inequality follows from Lemma 3.2.2 and the last inequality from the Lipschitz continuity of the gradient. Combining the last two relations, we obtain

$$\|e(x^i)\| \leq \frac{\eta_i L}{1 - \eta_i L} \|x^{i+1} - x^i\|.$$

It can be verified that conditions (3.2.1)-(3.2.3) are satisfied provided

$$\eta_i < \frac{1}{2L}.$$

3.4 Concluding Remarks

A unified approach to the analysis of perturbed feasible descent methods has been presented. It was established that a certain ε-approximate solution can be obtained where ε depends on the level of perturbations linearly. It is shown that the perturbed gradient projection, proximal minimization and extragradient methods fall within the presented framework. Applications of the ideas presented here to other classes of optimization algorithms (for example, [61, 40]) is an interesting subject of future research.

Chapter 4

Partially Asynchronous Inexact Parallel Variable Distribution Algorithms

We consider the recently proposed parallel variable distribution (PVD) algorithm [12] for solving optimization problems in which the variables are distributed among p processors. Each processor has the primary responsibility for updating its block of variables while allowing the remaining "secondary" variables to change in a restricted fashion along some easily computable directions. We propose a useful partially asynchronous approach and a generalization that consists of inexact subproblem solution in the PVD algorithm [64]. These modifications are the key features of the algorithm that has not been analyzed before. The proposed modified algorithms are more practical and make it easier to achieve good load balancing among the parallel processors. We present a general framework for the analysis of this class of algorithms and derive some new and improved linear convergence results for problems with weak sharp minima of order 2 and strongly convex problems. We also show that nonmonotone synchronization schemes are admissible, which further improves flexibility of PVD approach.

4.1 Introduction

We consider the general unconstrained optimization problem

$$\min_{x \in \Re^n} f(x), \tag{4.1.1}$$

where $f(\cdot) \in C^1_L(\Re^n)$. We first state the original PVD algorithm [12]. Let $x \in \Re^n$ be partitioned into p blocks $x_1, \ldots, x_p$, such that $x_l \in \Re^{n_l}$, $\sum_{l=1}^p n_l = n$. These blocks of variables are then distributed among p parallel processors. Each processor has the primary responsibility for updating its block of variables by solving a subproblem (see Algorithm 4.1.1 below) in which the remaining "secondary" variables are allowed to change in a restricted fashion along some easily computable directions. The distinctive novel feature of this algorithm is the presence of the "forget-me-not" term $x^i_{\bar{l}} + D^i_{\bar{l}}\mu_{\bar{l}}$ in the parallel subproblems (4.1.2). The presence of this term allows for a change in "secondary" variables. This makes PVD fundamentally different from the block Jacobi [3], coordinate descent [66] and parallel gradient distribution algorithms [33]. The directions

$D^i_{\bar{l}}$ are typically easily computable steepest descent or quasi-Newton directions in the space of the corresponding variables. The "forget-me-not" approach improves robustness and accelerates convergence of the algorithm and is the key to its success. The parallelization phase is followed by a simple synchronization step which picks up a point with the objective function value at least as good as the smallest among all the new points computed by the parallel processors.

Algorithm 4.1.1 (PVD) *Start with any* $x^0 \in \Re^n$. *Having* x^i, *stop if* $\nabla f(x^i) = 0$. *Otherwise, compute* x^{i+1} *as follows :*

(•) Parallelization : *For each processor* $l \in \{1, \ldots, p\}$ *compute*

$$(y^i_l, \mu^i_{\bar{l}}) \in \arg \min_{x_l, \mu_{\bar{l}}} \psi^i_l(x_l, \mu_{\bar{l}}) := f(x_l, x^i_{\bar{l}} + D^i_{\bar{l}} \mu_{\bar{l}}). \tag{4.1.2}$$

(•) Synchronization : *Compute* x^{i+1} *such that*

$$f(x^{i+1}) \leq \min_{l \in \{1 \ldots, p\}} \psi^i_l(y^i_l, \mu^i_{\bar{l}}). \tag{4.1.3}$$

We will sometimes refer to x^i as the base point at the $(i+1)$-st iteration. In the above algorithm $\bar{l}$ denotes the complement of l in the set $\{1, \ldots, p\}$ and $\mu_{\bar{l}} \in \Re^{p-1}$. The matrix $D^i_{\bar{l}}$ is an $n_{\bar{l}} \times (p-1)$ block diagonal matrix formed by placing the blocks $d^i_1, \ldots, d^i_{p-1}$ ($d^i_t \in \Re^{n_t}$, $t = 1, \ldots, p-1$) of an arbitrary direction $d^i \in \Re^n$ along its block diagonal as follows :

$$D_l^i := \begin{pmatrix} d_1^i & & & & & & \\ & d_2^i & & & & & \\ & & \ddots & & & & \\ & & & d_{l-1}^i & & & \\ & & & & d_{l+1}^i & & \\ & & & & & \ddots & \\ & & & & & & d_p^i \end{pmatrix}$$

In [12], the proposed synchronization step consists of minimizing the objective function in the affine hull of all the points computed in parallel by the p processors.

In [12] it was shown that every accumulation point of the PVD iterates is a stationary point of $f(\cdot)$ if an *exact global solution* to subproblems (4.1.2) is computed at every iteration. It was also established that, in the strongly convex case, the iterates converge to the problem solution at a linear rate.

We point out that the *global solution* requirement in the general (nonconvex) case is impractical. In Section 4.3 we show that it is possible to get rid of this requirement by imposing a certain sufficient descent condition instead. Section 4.3 also covers the partially asynchronous approach and contains some new convergence results for problems with weak sharp minima of order 2. We note that the original requirement of *exact subproblem solution* is also undesirable. In Section 4.2 we describe an algorithm with inexact subproblem solution in the convex case and derive a sharper linear convergence result than the one given in [12]. We emphasize that the partially asynchronous and inexact subproblem solution approaches provide a flexible framework that allows for effective load balancing among the parallel processors. In Section 4.3 we also exhibit that synchronization step can be combined with nonmonotone stabilization schemes, if needed.

4.2 PVD with inexact subproblem solution

In this section we propose a computationally important modification of the PVD algorithm in which the subproblems (4.1.2) in the Algorithm 4.1.1 are solved approximately. It is clear that in practice insisting on exact solution of those subproblems is undesirable, and often unrealistic. Even when it is possible to compute these solutions accurately, it can be wasteful doing so, especially in the initial stages of the minimization process.

Our results show that there is no need to wait until exact solutions to all the subproblems are found (which can result in considerable idle times for processors that have already completed their work). Instead, we can accept the current approximations to solutions of the subproblems and proceed to the synchronization step, provided those approximations are reasonably good. This approach is more robust and allows for flexible synchronization schemes thus making it easier to achieve good load balancing among the parallel processors. In particular, we show that we can solve the subproblems to within $\varepsilon(\cdot)$-*stationarity* (see

Section 2.1), and yet guarantee the linear convergence rate if $f(\cdot)$ is strongly convex. The tolerance for an l-th parallel subproblem depends linearly on the the norm of the corresponding portion of the gradient at the current base point (see (4.2.2) and (4.2.7)).

By making an explicit use of the "forget-me-not" terms in the subproblems, we also improve on the linear convergence result given in [12]. In [12] it is established that, for the strongly convex case, the following estimate is valid

$$\|x^i - \bar{x}\| \leq c_1 \left(1 - \frac{c_2}{p}\right)^{\frac{i}{2}},$$

where $\bar{x}$ is the (unique) solution of the problem, p is the number of parallel processors, and c_1, c_2 are positive constants. This result is not quite satisfactory because the presense of p in the denominator suggests that the convergence speed goes down as the number of processors used increases. We point out that the proof given in [12] fails to make use of the "forget-me-not" terms which are the key to the algorithm. By refining the proof, we obtain a better convergence speed estimate

$$\|x^i - \bar{x}\| \leq c_1 (1 - c_3)^{\frac{i}{2}},$$

where $c_3 > 0$ does not depend on p. Therefore convergence speed of the algorithm does not deteriorate as the number of processors used increases, provided certain natural conditions are imposed on the "forget-me-not" terms.

We consider the following algorithm.

Algorithm 4.2.1 *Start with any $x^0 \in \Re^n$. Having x^i, stop if $\nabla f(x^i) = 0$. Otherwise, compute x^{i+1} as follows :*

(•) Parallelization : *for each processor* $l \in \{1, \ldots, p\}$ *compute* $(y_l^i, \mu_{\bar{l}}^i)$ *as an* $\varepsilon_{i,l}$*-approximate solution (see (4.2.2)) of*

$$\min_{x_l, \mu_{\bar{l}}} \psi_l^i(x_l, \mu_{\bar{l}}) := f(x_l, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}).$$

(•) Synchronization : *Compute* x^{i+1} *such that*

$$f(x^{i+1}) \leq \min_{l \in \{1 \ldots, p\}} \psi_l^i(y_l^i, \mu_{\bar{l}}^i). \tag{4.2.1}$$

To make the parallelization step precise, we say that the current approximation to the solution of a subproblem is admissible if it belongs to an $\varepsilon(\cdot)$-stationary set of this subproblem (see Section 2.1). The parallelization subproblems are therefore equivalent to computing a point

$$(y_l^i, \mu_{\bar{l}}^i) \in X_s^{l,i}(\varepsilon_{i,l}) := \{(x_l, \mu_{\bar{l}}) \in \Re^{n_l + p - 1} \mid \|\nabla \psi_l^i(x_l, \mu_{\bar{l}})\| \leq \varepsilon_{i,l}\}. \tag{4.2.2}$$

We first establish some preliminary results. Let A_l^i be an $n \times (n_l + p - 1)$ matrix defined by

$$A_l^i = \begin{pmatrix} I_l & 0 \\ 0 & D_{\bar{l}}^i \end{pmatrix},$$

where I_l is an $n_l \times n_l$ identity matrix. We assume that every block d_t^i of $D_{\bar{l}}^i$ is normalized, that is $\|d_t^i\| = 1, \ t = 1, \ldots, p$. Then for any $y \in \Re^{n_l + p - 1}$ we have

$$\begin{aligned} \|A_l^i y\|^2 &= \sum_{j=1}^{n_l} y_j^2 + \sum_{j=n_l+1}^{n_l+p-1} y_j^2 \|d_j^i\|^2 \\ &= \sum_{j=1}^{n_l+p-1} y_j^2 \\ &= \|y\|^2, \end{aligned} \tag{4.2.3}$$

where the first equality follows from the block diagonal structure of $D^i_{\bar{l}}$. Hence $\|A^i_l\| = \|(A^i_l)^\top\| = 1$.

Lemma 4.2.1 *If $f(\cdot) \in C^1_L(\Re^n)$ then $\psi^i_l(\cdot,\cdot) \in C^1_L(\Re^{n_l+p-1})$ for any $i = 0, 1, \ldots$ and $l = 1, \ldots, p$.*

Proof. Note that

$$\begin{aligned}
\nabla\psi^i_l(x_l, \mu_{\bar{l}}) &= \begin{pmatrix} \nabla_l f(x_l, x^i_{\bar{l}} + \mu_{\bar{l}} D^i_{\bar{l}}) \\ (D^i_{\bar{l}})^\top \nabla_{\bar{l}} f(x_l, x^i_{\bar{l}} + \mu_{\bar{l}} D^i_{\bar{l}}) \end{pmatrix} \\
&= (A^i_l)^\top \nabla f(x_l, x^i_{\bar{l}} + D^i_{\bar{l}} \mu_{\bar{l}})
\end{aligned} \tag{4.2.4}$$

For any $(x_l, \mu_{\bar{l}}), (z_l, \nu_{\bar{l}}) \in \Re^{n_l+p-1}$ we have

$$\begin{aligned}
\|\nabla\psi^i_l(x_l, \mu_{\bar{l}}) - \nabla\psi^i_l(z_l, \nu_{\bar{l}})\| &= \|(A^i_l)^\top \left(\nabla f(x_l, x^i_{\bar{l}} + D^i_{\bar{l}}\mu_{\bar{l}}) - \nabla f(z_l, x^i_{\bar{l}} + D^i_{\bar{l}}\nu_{\bar{l}})\right)\| \\
&\le \|(A^i_l)^\top\| \|\nabla f(x_l, x^i_{\bar{l}} + D^i_{\bar{l}}\mu_{\bar{l}}) - \nabla f(z_l, x^i_{\bar{l}} + D^i_{\bar{l}}\nu_{\bar{l}})\| \\
&\le L \left\| A^i_l \begin{pmatrix} x_l - z_l \\ \mu_{\bar{l}} - \nu_{\bar{l}} \end{pmatrix} \right\| \\
&= L\|(x_l, \mu_{\bar{l}}) - (z_l, \nu_{\bar{l}})\|,
\end{aligned}$$

where the second inequality follows from the fact that $\|(A^i_l)^\top\| = 1$, and $f(\cdot) \in C^1_L(\Re^n)$; the last equality follows from (4.2.3). We thus established that $\psi^i_l(\cdot,\cdot) \in C^1_L(\Re^{n_l+p-1})$, for all $l = 1, \ldots, p,\ i = 0, 1, \ldots$. ■

Lemma 4.2.2 *If $f(\cdot)$ is strongly convex with modulus $\theta > 0$ then $\psi^i_l(\cdot,\cdot)$ is strongly convex with modulus $\theta > 0$ for any $i = 0, 1, \ldots$ and $l = 1, \ldots, p$.*

Proof. Making use of (4.2.4), we have

$$
\begin{aligned}
& (\nabla \psi_l^i(x_l, \mu_{\bar{l}}) - \nabla \psi_l^i(z_l, \nu_{\bar{l}}))^\top ((x_l, \mu_{\bar{l}}) - (z_l, \nu_{\bar{l}})) \\
= \; & \left((A_l^i)^\top (\nabla f(x_l, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}) - \nabla f(z_l, x_{\bar{l}}^i + D_{\bar{l}}^i \nu_{\bar{l}})) \right)^\top \begin{pmatrix} x_l - z_l \\ \mu_{\bar{l}} - \nu_{\bar{l}} \end{pmatrix} \\
= \; & \left(\nabla f(x_l, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}) - \nabla f(z_l, x_{\bar{l}}^i + D_{\bar{l}}^i \nu_{\bar{l}}) \right)^\top A_l^i \begin{pmatrix} x_l - z_l \\ \mu_{\bar{l}} - \nu_{\bar{l}} \end{pmatrix} \\
= \; & \left(\nabla f(x_l, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}) - \nabla f(z_l, x_{\bar{l}}^i + D_{\bar{l}}^i \nu_{\bar{l}}) \right)^\top \begin{pmatrix} x_l - z_l \\ D_{\bar{l}}^i(\mu_{\bar{l}} - \nu_{\bar{l}}) \end{pmatrix} \\
\geq \; & \theta \left\| \begin{pmatrix} x_l - z_l \\ D_{\bar{l}}^i(\mu_{\bar{l}} - \nu_{\bar{l}}) \end{pmatrix} \right\|^2 \\
= \; & \theta \left\| A_l^i \begin{pmatrix} x_l - z_l \\ \mu_{\bar{l}} - \nu_{\bar{l}} \end{pmatrix} \right\|^2 \\
= \; & \theta \| (x_l, \mu_{\bar{l}}) - (z_l, \nu_{\bar{l}}) \|^2,
\end{aligned}
$$

where the inequality follows from strong convexity of $f(\cdot)$, and the last equality follows from (4.2.3). Hence $\psi_l^i(\cdot, \cdot)$ is strongly convex with modulus θ. ∎

For simplicity of presentation, from now on we assume that

$$
d_t^i = \frac{\nabla_t f(x^i)}{\|\nabla_t f(x^i)\|}, \quad t = 1, \ldots, p.
$$

For this choice of directions, we have

$$(A_l^i)^\top \nabla f(x^i) = \begin{pmatrix} \nabla_l f(x^i) \\ \langle d_1^i, \nabla_1 f(x^i)\rangle \\ \vdots \\ \langle d_{l-1}^i, \nabla_{l-1} f(x^i)\rangle \\ \langle d_{l+1}^i, \nabla_{l+1} f(x^i)\rangle \\ \vdots \\ \langle d_p^i, \nabla_p f(x^i)\rangle \end{pmatrix} = \begin{pmatrix} \nabla_l f(x^i) \\ \|\nabla_1 f(x^i)\| \\ \vdots \\ \|\nabla_{l-1} f(x^i)\| \\ \|\nabla_{l+1} f(x^i)\| \\ \vdots \\ \|\nabla_p f(x^i)\| \end{pmatrix}.$$

Hence, by (4.2.4),

$$\begin{aligned} \|\nabla \psi_l^i(x_l^i, 0)\| &= \|(A_l^i)^\top \nabla f(x^i)\| \\ &= \|\nabla f(x^i)\|. \end{aligned} \tag{4.2.5}$$

The latter property enables us to explicitly relate solutions of the parallel subproblems (4.1.2) to the progress being made towards solving the original problem (4.1.1). This is the key to our generalizations as well as improved convergence results.

We note that instead of the scaled gradient directions we could take any other directions satisfying the natural conditions

$$|\langle d_t^i, \nabla_t f(x^i)\rangle| \geq \sigma_t(\|\nabla_t f(x^i)\|), \quad t = 1, \ldots, p,$$

where $\sigma_t(\cdot)$ are forcing functions (see Definition 1.2.1). Depending on the particular forcing functions, some arguments in the subsequent analysis may need to be changed.

We are now ready to prove our main results.

Theorem 4.2.1 *Suppose $f(\cdot)$ is strongly convex with modulus $\theta > 0$ and $f(\cdot) \in C_L^1(\Re^n)$. If*

$$\sum_{i=0}^{\infty} \max_{l \in \{1,\ldots,p\}} \varepsilon_{i,l}^2 < \infty, \tag{4.2.6}$$

then every sequence $\{x^i\}$ generated by Algorithm 4.2.1 converges to the solution $\bar{x}$ of (4.1.1). Moreover, if

$$\varepsilon_{i,l} \leq \beta \|\nabla_l f(x^i)\|, \quad 0 \leq \beta < \sqrt{\frac{\theta}{L}} \tag{4.2.7}$$

then $\{x^i\}$ converges to $\bar{x}$ R-linearly :

$$\|x^i - \bar{x}\| \leq \left(\frac{2}{\theta}(f(x^0) - f(\bar{x}))\right)^{\frac{1}{2}} \left(1 - \frac{\theta(\theta - L\beta^2)}{L^2}\right)^{\frac{i}{2}}.$$

Proof. For any iteration $i = 0, 1, \ldots$ and any processor $l = 1, \ldots, p$, by (4.2.2) and Lemma 2.4.2, we have that

$$\psi_l^i(y_l^i, \mu_l^i) \leq \bar{\psi}_l^i + \frac{\varepsilon_{i,l}^2}{2\theta}, \tag{4.2.8}$$

where $\bar{\psi}_l^i$ is the exact optimal value of the corresponding subproblem. Define an auxiliary point

$$\Re^{n_l+p-1} \ni (z_l^i, \nu_l^i) := (x_l^i, 0) - \frac{1}{L} \nabla \psi_l^i(x_l^i, 0).$$

We further obtain

$$
\begin{aligned}
f(x^i) - f(y_l^i, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}^i) &= \psi_l^i(x_l^i, 0) - \psi_l^i(y_l^i, \mu_{\bar{l}}^i) \\
&\geq \psi_l^i(x_l^i, 0) - \bar{\psi}_l^i - \frac{\varepsilon_{i,l}^2}{2\theta} \\
&\geq \psi_l^i(x_l^i, 0) - \psi_l^i(z_l^i, \nu_{\bar{l}}^i) - \frac{\varepsilon_{i,l}^2}{2\theta} \\
&\geq \frac{1}{2L}\|\nabla \psi_l^i(x_l^i, 0)\|^2 - \frac{\varepsilon_{i,l}^2}{2\theta} \\
&= \frac{1}{2L}\|\nabla f(x^i)\|^2 - \frac{\varepsilon_{i,l}^2}{2\theta},
\end{aligned}
\tag{4.2.9}
$$

where the first inequality follows from (4.2.8), the third inequality from Lemma 1.2.1, and the last equality from (4.2.5). By (4.2.1), we have

$$
\begin{aligned}
f(x^i) - f(x^{i+1}) &\geq f(x^i) - f(y_l^i, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}^i) \\
&\geq \frac{1}{2L}\|\nabla f(x^i)\|^2 - \frac{1}{2\theta} \max_{l \in \{1,\ldots,p\}} \varepsilon_{i,l}^2.
\end{aligned}
\tag{4.2.10}
$$

From (4.2.10) we have

$$
f(x^{i+1}) \leq f(x^i) + \frac{1}{2\theta} \max_{l \in \{1,\ldots,p\}} \varepsilon_{i,l}^2.
$$

Note that, by strong convexity of $f(\cdot)$, the sequence $\{f(x^i)\}$ is bounded below. Hence, by Lemma 1.2.2 and (4.2.6), it follows that the sequence $\{f(x^i)\}$ converges. Therefore $\{f(x^i) - f(x^{i+1})\} \to 0$. Since, by (4.2.6),

$$
\lim_{i \to \infty} \max_{l \in \{1,\ldots,p\}} \varepsilon_{i,l}^2 = 0,
$$

we conclude from (4.2.10) that $\{\|\nabla f(x^i)\|\} \to 0$. Since $\bar{x}$, the solution of (4.1.1), is the unique stationary point, it follows that $\{x^i\}$ converges to $\bar{x}$.

If (4.2.7) holds, then from (4.2.9) we obtain

$$\begin{aligned} f(x^i) - f(x^{i+1}) &\geq \frac{1}{2L}\|\nabla f(x^i)\|^2 - \frac{\beta^2}{2\theta}\|\nabla_l f(x^i)\|^2 \\ &\geq \frac{1}{2L}\|\nabla f(x^i)\|^2 - \frac{\beta^2}{2\theta}\|\nabla f(x^i)\|^2 \\ &= \frac{\theta - L\beta^2}{2L\theta}\|\nabla f(x^i)\|^2, \end{aligned} \tag{4.2.11}$$

where the second inequality follows from monotonicity of the 2-norm. Note that by (4.2.7), $\frac{\theta - L\beta^2}{2L\theta} > 0$. The rest of the proof is standard. By Lemma 1.2.1, it follows that

$$\begin{aligned} \frac{L}{2}\|x^i - \bar{x}\|^2 &\geq f(x^i) - f(\bar{x}) - \langle \nabla f(\bar{x}), x^i - \bar{x} \rangle \\ &= f(x^i) - f(\bar{x}) \end{aligned} \tag{4.2.12}$$

By the Cauchy-Schwartz inequality and strong convexity of $f(\cdot)$, it follows that

$$\begin{aligned} \|\nabla f(x^i)\|\|x^i - \bar{x}\| &= \|\nabla f(x^i) - \nabla f(\bar{x})\|\|x^i - \bar{x}\| \\ &\geq \langle \nabla f(x^i) - \nabla f(\bar{x}), x^i - \bar{x} \rangle \\ &\geq \theta\|x^i - \bar{x}\|^2. \end{aligned}$$

Hence

$$\|\nabla f(x^i)\| \geq \theta\|x^i - \bar{x}\|.$$

Combining the last inequality with (4.2.11), we obtain

$$f(x^i) - f(x^{i+1}) \geq \frac{\theta(\theta - L\beta^2)}{2L}\|x^i - \bar{x}\|^2.$$

This together with (4.2.12) yields

$$f(x^i) - f(x^{i+1}) \geq \frac{\theta(\theta - L\beta^2)}{L^2}(f(x^i) - f(\bar{x})).$$

Rearranging terms gives

$$f(x^{i+1}) - f(\bar{x}) \le \left(1 - \frac{\theta(\theta - L\beta^2)}{L^2}\right)(f(x^i) - f(\bar{x})).$$

Hence the sequence $\{f(x^i)\}$ converges Q-linearly. Successive application of the last inequality yields

$$f(x^i) - f(\bar{x}) \le \left(1 - \frac{\theta(\theta - L\beta^2)}{L^2}\right)^i (f(x^0) - f(\bar{x})).$$

By strong convexity of $f(\cdot)$, we have

$$\begin{aligned} \frac{\theta}{2}\|x^i - \bar{x}\|^2 &\le f(x^i) - f(\bar{x}) - \langle \nabla f(\bar{x}), x^i - \bar{x} \rangle \\ &= f(x^i) - f(\bar{x}). \end{aligned}$$

Hence the sequence $\{x^i\}$ converges R-linearly. In particular, we have

$$\|x^i - \bar{x}\| \le \left(\frac{2}{\theta}(f(x^i) - f(\bar{x}))\right)^{\frac{1}{2}},$$

and

$$\|x^i - \bar{x}\| \le \left(\frac{2}{\theta}(f(x^0) - f(\bar{x}))\right)^{\frac{1}{2}} \left(1 - \frac{\theta(\theta - L\beta^2)}{L^2}\right)^{\frac{i}{2}}.$$

■

For the convex case, we have the following result.

Theorem 4.2.2 *Suppose $f(\cdot)$ is convex and $f(\cdot) \in C^1_L(\Re^n)$. Let $\mathcal{L}(f, x^0) := \{x \mid f(x) \le f(x^0)\}$ and $B = \{x \mid \|x\| \le 1\}$. Suppose $\mathcal{L}(f, x^0) \subset x^0 + rB$, $\quad r > 0$. If*

$$\varepsilon_{i,l} \le \beta \|\nabla_l f(x^i)\|^2, \quad 0 \le \beta < \frac{1}{2Lr},$$

or

$$\sum_{i=0}^{\infty} \max_{l \in \{1,\ldots,p\}} \varepsilon_{i,l} < \infty,$$

then every accumulation point of any sequence $\{x^i\}$ generated by Algorithm 4.2.1 is a solution of (4.1.1).

Proof. First note that under our assumptions $\mathcal{L}(f, x^0)$ is bounded and hence X_{opt} is nonempty. Furthermore, for all i

$$d(x^i, X_{opt}) \leq r.$$

Apllying Lemma 2.4.2, similarly to the proof of Theorem 4.2.1, we obtain

$$f(x^i) - f(x^{i+1}) \geq \frac{1}{2L}\|\nabla f(x^i)\|^2 - r\varepsilon_{i,l}.$$

The rest of the proof can be patterned after that of Theorem 4.2.1. ∎

4.3 Partially Asynchronous PVD

In this section, we present a practical version of the PVD algorithm for the general (nonconvex) case. In particular, we show that there is no need to find an exact global solution for the subproblems. Any point that satisfies a natural sufficient descent condition can be accepted for the next iteration. We note, in the passing, that the proof given in [12] makes use of exact global solutions in an essential way and breaks down if, for example, only stationary points in the subproblems are available. We further point out that partial asynchronization of the p parallel processors is possible by allowing each of the p processors to take as many steps as desired by individually updating its base point. Synchronization can be performed at any time provided every processor has achieved the sufficient descent condition. Furthermore, we show that synchronization step need not be monotone and can be combined with nonmonotone stabilization schemes similar to [18].

We also derive some new convergence results for weakly sharp problems of order 2 (see Definition 4.3.1). This class of problems can be viewed as a generalization of strongly convex problems and a certain unconstrained smooth analogue of weak sharp minima [5].

We begin by imposing a natural sufficient descent condition on an algorithm (Algorithm A below) used to solve the subproblems (4.1.2) generated by the PVD Algorithm 4.1.1.

Algorithm A. *Given any function* $\varphi(\cdot) \in C^1_L(\Re^m)$ *and any starting point* $t^0 \in \Re^m$ *generate a point* $t^* \in \Re^m$ *such that*

$$\varphi(t^0) \geq \varphi(t^*) + \gamma \|\nabla \varphi(t^0)\|^2, \tag{4.3.1}$$

where $\gamma > 0$ *depends on* L *and does not depend on* t^0*.*

Note that the above condition is satisfied by a single iteration of any reasonable descent algorithm [33] applied to the problem of minimizing $\varphi(\cdot)$ with t^0 as a starting point. Hence it is also satisfied for a minimum or a stationary point computed by some descent algorithm provided it uses t^0 as a starting point.

We now state our partially asynchronous PVD algorithm.

Algorithm 4.3.1 *Start with any* $x^0 \in \Re^n$. *Having* x^i, *stop if* $\nabla f(x^i) = 0$. *Otherwise, compute* x^{i+1} *as follows :*

(•) Parallelization : *for each processor* $l \in \{1, \ldots, p\}$ *generate* $(y_l^i, \mu_{\bar{l}}^i)$ *by applying Algorithm A one or more times to the problem*

$$\min_{x_l, \mu_{\bar{l}}} \psi_l^i(x_l, \mu_{\bar{l}}) := f(x_l, x_{\bar{l}}^i + D_{\bar{l}}^i \mu_{\bar{l}}) \tag{4.3.2}$$

using $(x_l^i, 0)$ *as a starting point.*

(•) Synchronization : *Compute* x^{i+1} *such that*

$$f(x^{i+1}) \leq \max_{l \in \{1 \ldots, p\}} \psi_l^i(y_l^i, \mu_{\bar{l}}^i) + \lambda\gamma \|\nabla f(x^i)\|^2, \tag{4.3.3}$$

where $\lambda \in (0, 1)$.

Note that once the sufficient descent condition (4.3.1) with respect to $f(x^i) = \psi_l^i(x_l^i, 0)$ is satisfied, each processor can independently update its base point, generate new directions $D_{\bar{l}}^i$ and proceed to find a point with better objective function value. After these parallel steps are performed by each processor then an eventual synchronization step is taken. Note that our synchronization step may increase rather than decrease the objective function when compared to the values obtained by the parallel processors. This provides the algorithm with more flexibility and is known to be useful in nonlinear nonconvex optimization [17, 18].

We next introduce a notion of weak sharp minima of order 2 which allows us to strengthen some of the traditional convergence results.

Definition 4.3.1 *We say that a set of (local) minima X_s is weakly sharp of order 2 if there exist positive constants ρ and ϵ such that*

$$f(x) - f([x]^+) \geq \rho d(x, X_s)^2 \quad \forall x \in X_s + \epsilon B, \tag{4.3.4}$$

where $[\cdot]^+$ denotes the orthogonal projection map onto X_s, $d(\cdot, X_s)$ denotes the 2-norm distance to the set X_s, and $B = \{x \in \Re^n \mid \|x\| \leq 1\}$ is the closed unit ball in $\Re^n$.

The class of problems with weak sharp minima of order 2 can be thought of as a certain unconstrained smooth analogue of weak sharp minima (of order 1) [38, 51, 5]. Note that it subsumes strongly convex programs. Let $f(\cdot)$ be strongly convex with modulus 2ρ. Then its unique optimal point $\bar{x}$ is globally (with $\epsilon = \infty$) weakly sharp of order 2. This can be easily verified as follows. By strong convexity, for any $x \in \Re^n$

$$\begin{aligned} f(x) - f(\bar{x}) &\geq \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{2\rho}{2}\|x - \bar{x}\|^2 \\ &= \rho\|x - \bar{x}\|^2 \\ &= \rho d(x, X_s)^2. \end{aligned}$$

Hence the growth property of $f(\cdot)$ (near the solution set) in the Definition 4.3.1 is a generalization of strong convexity. It is clear that there exist functions with weak sharp minima of order 2 which are not strongly convex (or even convex) in any neighborhood of their solution sets. One example is

$$f(x) := (x_1^2 + x_2^2 - 1)^2, \quad x \in \Re^2.$$

The stationary set of this function is

$$X_s = \{x \mid x_1^2 + x_2^2 = 1\} \cup \{(0,0)\} := X_s^1 \cup X_s^2$$

with X_s^1 being the set of minima. It is easy to see that X_s^1 is a set of weak sharp minima of order 2 (with $\rho = 1$ and $\epsilon = 1/2$). Indeed, for any $x \in X_s^1 + \frac{1}{2}B$

$$\begin{aligned} d(x, X_s^1)^2 &= |1 - (x_1^2 + x_2^2)|^2 \\ &= (x_1^2 + x_2^2 - 1)^2 \\ &= f(x) - f(\bar{x}) \quad \forall \bar{x} \in X_s^1. \end{aligned}$$

Obviously, even locally (in any neighborhood of X_s^1) $f(\cdot)$ in this example is neither strongly convex nor convex. However, we are able to strengthen standard convergence results for problems of this class (see Theorem 4.3.1 below). As an aside, we note that $X_s^2 = \{(0,0)\}$ is a set of weak sharp maxima in the sense of the same definition (with the sign of the left-hand-side of (4.3.4) reversed).

Remark 4.3.2 contains further examples of problems with weak sharp minima of order 2. Some issues related to this growth condition are also discussed in [77].

Theorem 4.3.1 *Let $f(\cdot) \in C_L^1(\Re^n)$. Suppose $\{x^i\}$ is any sequence generated by Algorithm 4.3.1. Then either $f(\cdot)$ is unbounded from below on $\Re^n$ or the sequence $\{f(x^i)\}$ converges, the sequence $\{\nabla f(x^i)\}$ converges to zero and for every accumulation point $\bar{x}$ of the sequence $\{x^i\}$ it follows that $\nabla f(\bar{x}) = 0$.*

Suppose the subset X_s of stationary points of $f(\cdot)$ that contains accumulation points of $\{x^i\}$ is a set of weak sharp minima of order 2. Then the sequence

$\{f(x^i)\}$ converges Q-linearly, and the sequences $\{\nabla f(x^i)\}$ and $\{d(x^i, X_s)\}$ converge to zero R-linearly.

Proof. By Lemma 4.2.1, for any iteration $i = 0, 1, \ldots$ and any processor $l = 1, \ldots, p, \quad \psi_l^i(\cdot,\cdot) \in C_L^1(\Re^{n_l+p-1})$ (with the same L). By (4.3.1) and (4.2.5), it follows that

$$\begin{aligned} \psi_l^i(x_l^i, 0) - \psi_l^i(y_l^i, \mu_l^i) &\geq \gamma \|\nabla \psi_l^i(x_l^i, 0)\|^2 \\ &= \gamma \|\nabla f(x^i)\|^2, \end{aligned}$$

Since the last inequality holds for all $l = 1, \ldots, p$, we have

$$f(x^i) - \max_{l \in \{1,\ldots,p\}} \psi_l^i(y_l^i, \mu_l^i) \geq \gamma \|\nabla f(x^i)\|^2.$$

Hence, by the synchronization step (4.3.3),

$$f(x^i) - \left(f(x^{i+1}) - \lambda\gamma \|\nabla f(x^i)\|^2 \right) \geq \gamma \|\nabla f(x^i)\|^2$$

and

$$f(x^i) - f(x^{i+1}) \geq (1-\lambda)\gamma \|\nabla f(x^i)\|^2. \tag{4.3.5}$$

We immediately conclude that $\{f(x^i)\}$ is a monotonically nonincreasing sequence. If this sequence is bounded from below then it converges. In the latter case, $\{f(x^i) - f(x^{i+1})\} \to 0$ and consequently $\{\nabla f(x^i)\} \to 0$. Hence, by continuity of $\nabla f(\cdot)$, if there exist accumulation points of $\{x^i\}$, all of them are stationary points of $f(\cdot)$.

Suppose now the sequence $\{x^i\}$ has accumulation points. The preceding discussion immediately implies that the set of stationary points of $f(\cdot)$ is nonempty. Denote by X_s its subset that contains accumulation points of $\{x^i\}$. Clearly, $\{d(x^i, X_s)\} \to 0$. Hence $x^i \in X_s + \epsilon B$ for i sufficiently large, say $i \geq i_0$. Suppose X_s is weakly sharp of order 2. Then (4.3.4) is satisfied for all $i \geq i_0$.

By Lemma 1.2.1,

$$\begin{aligned} f(x^i) - f([x^i]^+) &\leq \langle \nabla f(x^i), x^i - [x^i]^+ \rangle + \frac{L}{2}\|x^i - [x^i]^+\|^2 \\ &= \langle \nabla f(x^i), x^i - [x^i]^+ \rangle + \frac{L}{2} d(x^i, X_s)^2, \end{aligned}$$

where $[\cdot]^+$ denotes the orthogonal projection onto X_s. Hence for all $i \geq i_0$, by (4.3.4), we obtain

$$\begin{aligned} \langle \nabla f(x^i), x^i - [x^i]^+ \rangle &\geq f(x^i) - f([x^i]^+) - \frac{L}{2} d(x^i, X_s)^2 \\ &\geq f(x^i) - f([x^i]^+) - \frac{L}{2\rho}(f(x^i) - f([x^i]^+)) \\ &= \left(1 - \frac{L}{2\rho}\right)(f(x^i) - f([x^i]^+)), \end{aligned} \tag{4.3.6}$$

By the Cauchy-Schwartz inequality and (4.3.4), we further obtain

$$\begin{aligned} \|\nabla f(x^i)\| d(x^i, X_s) &\geq \langle \nabla f(x^i), x^i - [x^i]^+ \rangle \\ &\geq \left(1 - \frac{L}{2\rho}\right)(f(x^i) - f([x^i]^+)) \\ &\geq \rho\left(1 - \frac{L}{2\rho}\right) d(x^i, X_s)^2. \end{aligned}$$

Hence

$$\|\nabla f(x^i)\| \geq \rho\left(1 - \frac{L}{2\rho}\right) d(x^i, X_s). \tag{4.3.7}$$

By (4.3.7), the Cauchy-Schwartz inequality and (4.3.6) we have

$$\begin{aligned}
\|\nabla f(x^i)\|^2 &\geq \|\nabla f(x^i)\|\rho\left(1-\frac{L}{2\rho}\right)d(x^i,X_s) \\
&\geq \rho\left(1-\frac{L}{2\rho}\right)\langle\nabla f(x^i), x^i-[x^i]^+\rangle \\
&\geq \rho\left(1-\frac{L}{2\rho}\right)^2(f(x^i)-f([x^i]^+)) \qquad (4.3.8)
\end{aligned}$$

Combining (4.3.5) and (4.3.8) gives

$$\begin{aligned}
f(x^i)-f(x^{i+1}) &\geq \gamma(1-\lambda)\|\nabla f(x^i)\|^2 \\
&\geq \gamma\rho(1-\lambda)\left(1-\frac{L}{2\rho}\right)^2(f(x^i)-f([x^i]^+)).
\end{aligned}$$

Rearranging terms, we obtain

$$f(x^{i+1})-f([x^i]^+) \leq \left(1-\gamma\rho(1-\lambda)\left(1-\frac{L}{2\rho}\right)^2\right)(f(x^i)-f([x^i]^+)).$$

We already established that the sequence $\{f(x^i)\}$ converges. Let $\bar{f} := \lim_{i\to\infty} f(x^i)$. Since all accumulation points of the sequence $\{x^i\}$ belong to the set X_s, it follows that accumulation points of the sequences $\{x^i\}$ and $\{[x^i]^+\}$ are the same. Therefore, by continuity of $f(\cdot)$, we obtain

$$\lim_{i\to\infty} f([x^i]^+) = \lim_{i\to\infty} f(x^i) = \bar{f}.$$

Because X_s is a set of (local) minima and $[x^i]^+ \in X_s$, it must be the case that $f([x^i]^+) = \bar{f}$ for all i sufficiently large, say $i \geq i_1$. Therefore, for $i \geq \max\{i_0, i_1\}$, we obtain

$$f(x^{i+1})-\bar{f} \leq \left(1-\gamma\rho(1-\lambda)\left(1-\frac{L}{2\rho}\right)^2\right)(f(x^i)-\bar{f}).$$

Hence the sequence $\{f(x^i)\}$ converges Q-linearly. By (4.3.5), the sequence $\{\nabla f(x^i)\}$ converges R-linearly to zero. Also, by (4.3.4), the sequence $\{d(x^i, X_s)\}$ converges R-linearly to zero. ∎

Remark 4.3.1 *At this time, it is an open question whether the sequence $\{x^i\}$ itself converges linearly under the assumptions of Theorem 4.3.1. Note that if we had a serial gradient descent method where*

$$x^{i+1} - x^i = -\eta_i \nabla f(x^i)$$

with the sequence of stepsizes $\{\eta_i\}$ uniformly bounded away from zero, then the linear convergence rate of $\{x^{i+1} - x^i\}$ (and hence also of $\{x^i\}$) would immediately follow from the linear convergence of $\{\nabla f(x^i)\}$. The difficulty with the PVD Algorithm is that we cannot explicitly relate $\{\nabla f(x^i)\}$ to $\{x^{i+1} - x^i\}$.

Careful re-examination of the proof of Theorem 4.3.1 shows that at the $(i+1)$-st iteration every parallel processor decreases the objective function $f(\cdot)$ of the original problem by a factor of $\|\nabla f(x^i)\|^2$ (this at least is true under our assumptions on the directions $d_t^i, \quad t = 1, \ldots, p$). Hence if the processors were to proceed with updating their base points completely independently without using any information from the other processors, we could still guarantee the same convergence results for each of the p sequences of iterates generated. Of course, this approach essentially yields p serial processes and therefore is a theoretical extreme. This observation is however of significance because it implies that we are allowed a lot of flexibility in devising partially asynchronous PVD algorithms. Also, as is evidenced by (4.3.3) and Theorem 4.3.1, we are not restricted to the "monotone" synchronization step proposed in the original PVD approach. It is known that always insisting on monotone decrease in the objective function at every iteration is not necessarily the best strategy [17]. We note that synchronization in PVD algorithms can be combined with nonmonotone stabilization schemes [18], provided the requirements of (4.3.3) are satisfied.

Remark 4.3.2 *A practically important example of weak sharp minima of order 2 is provided by the implicit Lagrangian reformulation [40] of the nonlinear complementarity problem.*

Consider the following nonlinear complementarity problem [8, 7] (NCP) of finding an $x \in \Re^n$ such that

$$F(x) \geq 0, \quad x \geq 0, \quad \langle x, F(x) \rangle = 0,$$

where $F : \Re^n \to \Re^n$ is a continuously differentiable mapping. In [40] it was established that the NCP can be solved via (smooth) unconstrained minimization of the following implicit Lagrangian function :

$$M(x,\alpha) := 2\alpha\langle x, F(x)\rangle + \|[x - \alpha F(x)]^+\|^2 - \|x\|^2 + \|[F(x) - \alpha x]^+\|^2 - \|F(x)\|^2,$$

where $\alpha > 1$ and $[\cdot]^+$ denotes the orthogonal projection onto the nonnegative orthant $\Re^n_+$. In particular, the implicit Lagrangian is nonnegative everywhere in $\Re^n$ and assumes the value of zero precisely at the solutions of the NCP.

In [29] it was established that

$$2(\alpha-1)\|r(x)\|^2 \le M(x,\alpha) \le 2\alpha(\alpha-1)\|r(x)\|^2, \quad \forall x \in \Re^n,$$

where $r(x) := x - [x - F(x)]^+$. Therefore the set of solutions X_s of the NCP is a set of weak sharp minima of order 2 for the implicit Lagrangian whenever the projection-type error bound holds :

$$d(x, X_s) \le \rho\|r(x)\| \quad \forall x \text{ with } \|r(x)\| \le \epsilon,$$

where ρ and ϵ are positive constants (independent of x). This error bound is known to hold when $F(\cdot)$ is affine (see [30, 53]) or $F(\cdot)$ has certain strong monotonicity structure (see [67, Theorem 2]). Moreover, under additional assumptions on $F(\cdot)$, this condition holds globally with $\epsilon = \infty$ (see [28, 29, 39, 48]).

Therefore our analysis shows that certain unconstrained optimization techniques applied to minimizing the implicit Lagrangian attain linear rate of convergence. This is a nice and nontrivial result given that the implicit Lagrangian is not known to be strongly convex in any neighborhood of its zero minima.

4.4 Concluding Remarks

Partially asynchronous parallel variable distribution algorithms and algorithms with inexact subproblem solution were proposed and analyzed. The modified algorithms present a flexible framework and make it easier to achieve good load balancing among the parallel processors. New and improved linear convergence results were derived for strongly convex problems and problems with weak sharp minima of order 2.

Bibliography

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice–Hall, Inc, Englewood, New Jersey, 1979.

[2] D.P. Bertsekas. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization*, May 1994. Submitted.

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

[4] P.T. Boggs and J.E. Dennis. A stability analysis for perturbed nonlinear iterative methods. *Mathematics of Computation*, 30:199–215, 1976.

[5] J.V. Burke and M.C. Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.

[6] F.H. Clarke. *Optimization and Nonsmooth Analysis.* John Wiley & Sons, New York, 1983.

[7] R.W. Cottle, F. Giannessi, and J.-L. Lions (editors). *Variational Inequalities and Complementarity Problems : Theory and Applications.* Whiley, New York, 1980.

[8] G.B. Dantzig and R.W. Cottle. Positive (semi-)definite programming. In J. Abadie, editor, *Nonlinear Programming*, pages 55–73, Amsterdam, 1967. North–Holland.

[9] C. Darken, J.Chang, and J.Moody. Learning rates schedules for faster stochastic gradient search. In *Neural Networks for signal processing 2*, New York, 1992. IEEE Press.

[10] C. Darken and J.Moody. Towards faster stochastic gradient search. In G. Tesauro J.D. Cowan and J. Alspector, editors, *Advances in Neural Information Processing Systems 4*, pages 1009–1016, San Francisco, CA, 1991. Morgan Kaufmann Publishers.

[11] P. A. Dorofeev. On some properties of quasi-gradient method. *USSR Computational Mathematics and Mathematical Physics*, 25:181–189, 1985.

[12] M.C. Ferris and O.L. Mangasarian. Parallel variable distribution. *SIAM Journal on Optimization*, 4(4):815–832, 1994.

[13] E.M. Gafni and D.P. Bertsekas. Two–metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, 22:936–964, 1984.

[14] D. Girard and H. Paugam-Moisy. Strategies for weight updating for parallel backpropagation. Technical Report 93-39, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, 46, Allée d'Italie, 69364 Lyon Cedex 07, France, 1993.

[15] A.A. Goldstein. Convex programming in Hilbert space. *Bull. Am. Math. Soc.*, 70:709–710, 1964.

[16] L. Grippo. A class of unconstrained minimization methods for neural network training. *Optimization Methods and Software*, 4:135–150, 1994.

[17] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal of Numerical Analysis*, 23:707–716, 1986.

[18] L. Grippo, F. Lampariello, and S. Lucidi. A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Mathematik*, 59:779–805, 1991.

[19] B. Hassibi and D.G. Stork. Optimal brain sergeon. In G. Tesauro J.D. Cowan and J. Alspector, editors, *Advances in Neural Information Processing Systems 5*, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[20] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, California, 1991.

[21] G. E. Hinton. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 1–12, Hillsdale, 1986. Erlbaum.

[22] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[23] M. Jabri and B. Flower. Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks. *IEEE Transactions on Neural Networks*, 3(1):154–157, 1992.

[24] T. Khanna. *Foundations of neural networks*. Addison–Wesley, New Jersey, 1989.

[25] G.M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[26] R. De Leone, E. Merelli, and R. Capparuccia. A modified backpropagation algorithm for neural network training, 1995.

[27] E.S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 1965.

[28] X.-D. Luo and P. Tseng. On global projection-type error bound for the linear complementarity problem. *Linear Algebra and Its Applications*. To appear.

[29] Z.-Q. Luo, O.L. Mangasarian, J. Ren, and M.V. Solodov. New error bounds for the linear complementarity problem. *Mathematics of Operations Research*, 19:880–892, 1994.

[30] Z.-Q. Luo and P. Tseng. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM Journal on Optimization*, 2:43–54, 1992.

[31] Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods : A general approach. *Annals of Operations Research*, 46:157–178, 1993.

[32] Z.-Q. Luo and P. Tseng. Analysis of an approximate gradient projection method with applications to the backpropagation algorithm. *Optimization Methods and Software*, 4:85–101, 1994.

[33] O.L. Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization*. To appear.

[34] O.L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969.

[35] O.L. Mangasarian. Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem. *SIAM Journal on Optimization*, 1:114–122, 1991.

[36] O.L. Mangasarian. Mathematical programming in neural networks. *ORSA Journal on Computing*, 5(4):349–360, 1993.

[37] O.L. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5(4):309–323, 1994.

[38] O.L. Mangasarian and R.R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

[39] O.L. Mangasarian and J. Ren. New improved error bounds for the linear complementarity problem. *Mathematical Programming*, 66:241–255, 1994.

[40] O.L. Mangasarian and M.V. Solodov. Nonlinear complementarity as unconstrained and constrained minimization. *Mathematical Programming*, 62:277–297, 1993.

[41] O.L. Mangasarian and M.V. Solodov. Backpropagation convergence via deterministic nonmonotone perturbed minimization. In G. Tesauro J.D. Cowan and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 383–390, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

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

[43] P. Marcotte. Application of Khobotov's algorithm to variational inequalities and network equilibrium problems. *Information Systems and Operational Research*, 29:258–270, 1991.

[44] B. Martinet. Regularisation d'inéquations variationelles per approximations successives. *Rev. Francaise d'Auto et.Inform. Rech. Opér.*, pages 154–159, 1970.

[45] D.Q. Mayne and E. Polak. Nondifferentiable optimization via adaptive smoothing. *Journal of Optimization Theory and Applications*, 43(4):601–614, 1984.

[46] V. S. Mikhalevitch, A. M. Gupal, and V. I. Norkin. *Methods of nonconvex optimization.* Nauka, Moscow, 1987. In Russian.

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

[48] J.-S. Pang. A posteriori error bounds for the linearly–constrained variational inequality problem. *Mathematics of Operations Research*, 12:474–484, 1987.

[49] H. Paugam-Moisy. On parallel algorithm for backpropagation by partitioning the training set. In *Neural Networks and Their Applications.* Proceedings of Fifth International Conference, Nimes, France, November 2-6, 1992.

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

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

[52] E. Rich and K. Knight. *Artificial Intelligence.* McGraw-Hill, New York, 1991.

[53] S.M. Robinson. Some continuity properties of polyhedral multifunctions. *Mathematical Programming Study*, 14:206–214, 1981.

[54] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[55] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[56] N. Rouche, P. Habets, and M Laloy. *Stability Theory by Liapunov's Direct Method.* Springer–Verlag, New York, 1977.

[57] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, pages 318–362, Cambridge, Massachusetts, 1986. MIT Press.

[58] T.J. Sejnowski and C.R. Rosenberg. Paralel networks that learn to pronounce english text. *Complex Systems*, 1:145–168, 1987.

[59] S. Shah, F. Palmieri, and M. Datum. Optimal filtering algorithms for fast learning in feedforward neural networks. *Neural Networks*, 5:779–787, 1992.

[60] J. Sietsma and R.J.F Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4:67–79, 1991.

[61] M. V. Solodov and P. Tseng. Modified pojection-type methods for monotone variational inequalities. Technical Report Mathematical Programming 94-04, Computer Science Department, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706, U.S.A., May 1994. SIAM Journal on Control and Optimization, to appear.

[62] M. V. Solodov and S. K. Zavriev. Stability properties of the gradient projection method with applications to the backpropagation algorithm. Technical Report Mathematical Programming 94-05, Computer Science Department, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706, U.S.A., June 1994. SIAM Journal on Optimization, submitted.

[63] M.V. Solodov. Convergence analysis of perturbed feasible descent methods. *Journal of Optimization Theory and Applications*, June 1995. Submitted.

[64] M.V. Solodov. Partially asynchronous inexact parallel variable distribution algorithms. *Computational Optimization and Applications*, June 1995. Submitted.

[65] A. Sperduti and A. Starita. Speed up learning and network optimization with extended backpropagation. *Neural Networks*, 6:365–383, 1993.

[66] P. Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical Programming*, 59:231–248, 1993.

[67] P. Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 1995. To appear.

[68] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, AC-31(9):803–812, 1986.

[69] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future:a connectionist approach. *International Journal of Neural Systems*, 1:193–209, 1990.

[70] H. White. Some asymptotic results for learning in single hidden-layer feedforward network models. *Journal of the American Statistical Association*, 84(408):1003–1013, 1989.

[71] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation. *IEEE Proceedings*, 78:1415–1442, September 1990.

[72] W.I. Zangwill. *Nonlinear Programming: A Unified Approach.* Prentice–Hall, Inc, Englewood Cliffs, New Jersey, 1969.

[73] S. K. Zavriev. Stochastic subgradient methods for Minmax problems. Izdatelstvo MGU, Moscow, 1984. In Russian.

[74] S. K. Zavriev. Convergence properties of the gradient method under variable level interference. *USSR Computational Mathematics and Mathematical Physics*, 30:997–1007, 1990.

[75] S.K. Zavriev and A.G. Perevozchikov. Attraction of trajectories of finite-difference inclusions and stability of numerical methods of stochastic non-smooth optimization. *Soviet Phys. Doklady*, 313:1373–1376, 1990.

[76] S.K. Zavriev and A.G. Perevozchikov. Direct Lyapunov's method in attraction analysis of finite-difference inclusions. *USSR Computational Mathematics and Mathematical Physics, Pergamon Press*, 30(1):22–32, 1990.

[77] R. Zhang and J. Treiman. Upper-Lipschitz multifunctions and inverse subdifferentials. *Nonlinear Analysis.* To appear.