Vol. 5 No. 1 JOURNAL OF COMPUTATIONAL MATHEMATICS January 1987

SOME RESULTS IN NONSMOOTH OPTIMIZATION*[1)]

YUAN YA-XIANG (袁亚湘)

(*Computing Centre, Academia Sinica, Beijing, China; Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England*)

Abstract

Some main results in nondifferentiable optimization are reviewed. In Section 2, we discuss subgradient methods. Section 3 is about the cutting plane method and the bundle methods are studied in Section 4. Trust region methods for composite nonsmooth optimization are discussed in Section 5.

§ 1. Introduction

A general nonsmooth optimization is to seek a point that attains the smallest value of a nonsmooth function $f(x)$, where $f(x)$ defined on $\mathbb{R}^n$ is continuous, but not necessarily differentiable. In other words, we need to solve the problem

$$\min f(x), \quad x\in\mathbb{R}^n. \tag{1.1}$$

A necessary condition for x^* to be a solution of (1.1) is that the null vector is in the subdifferential of $f(x)$ at x^*. The definition of the subdifferential can be found in Clark (1975), and is expressed as (2.6) in the next section.

Methods for solving (1.1), to be discussed in the next four sections, are all iterative. That means, to start calculation, an initial guess for the solution has to be made. Then at every iteration, a method would give a search direction or a trial step. In the first case, some kind of line search techniques are needed to choose a step-size $\alpha_k>0$ in order to move the approximate point from the original location (x_k, say) to a new one ($x_{k+1}=x_k+\alpha_k d_k$), where d_k is the search direction. In the second case, some kind of tests are needed to judge whether we should accept the trial step ($x_{k+1}=x_k+d_k$) or take a null step ($x_{k+1}=x_k$).

In the next section, we present some results of subgradient methods for (1.1). Most of the research in this area has been done by the Soviet scientists. In Section 3, we give a brief introduction to the cutting plane method. Section 4 is about the bundle methods, with an introduction of the conjugate subgradient method. In Section 5, we consider a class of trust region methods for the so-called composite optimization problem.

There are also many other methods that will not be discussed here. The readers are referred to Blinski and Wolfe (1975), Lemarechal and Mifflin (1978) and Nurminskii (1982).

* Received January 13, 1986.

1) Presented at the 50th Anniversary Conference of the Chinese Mathematical Society, December 1985, Shanghai.

§ 2. Subgradient Methods

For the moment, assume that the function $f(x)$ is continuously differentiable. The steepest descent method for solving (1.1) sets

$$x_{k+1}=x_k-\alpha_k\nabla f(x_k), \tag{2.1}$$

where $\alpha_k>0$ is a step–length. There are different techniques for choosing α_k, among which are the "exact line search" and the "Armijo type search". The former requires that

$$f(x_k-\alpha_k\nabla f(x_k))=\min_{t>0} f(x_k-t\nabla f(x_k)), \tag{2.2}$$

and the latter one chooses such α_k that satisfy the inequality

$$f(x_k-\alpha_k\nabla f(x_k))-f(x_k)\leqslant -c_1\alpha_k\|\nabla f(x_k)\|^2, \tag{2.3}$$

where c_1 is a parameter in (0, 1). For either search technique (2.2) or (2.3), it can be proved that any accumulation point of $\{x_k\}$ is a stationary point of (1.1), that is, the gradient of $f(x)$ is the null vector.

The subgradient method is a generalization of the steepest descent method (2.1). At every iteration, it lets

$$x_{k+1}=x_k-\alpha_k g_k, \tag{2.4}$$

where g_k is any subgradient of $f(x)$ at the point x_k. That is, we have

$$g_k\in\partial f(x_k), \tag{2.5}$$

where $\partial f(x_k)$ is subdifferential of $f(x)$ at x_k. The subdifferential $\partial f(x)$ of a function is defined by

$$\partial f(x)=\mathrm{conv}\{g\in\mathbb{R}^n\,|\,g=\lim_{i\to\infty}\nabla f(x_i),\ x_i\to x,\quad \nabla f(x_i)\text{ exist, }\nabla f(x_i)\text{ converge}\}. \tag{2.6}$$

For more details, see Clark (1975).

Hence a class of subgradient methods for solving (1.1) can be described below:

Algorithm **2.1** (The Subgradient Method).

Step 0: Given initial vector x_1.

Step 1: Calculate $f(x_k)$, and obtain a vector $g_k\in\partial f(x_k)$.

Step 2: Choose a step–size $\alpha_k>0$.

Step 3: Set

$$x_{k+1}=x_k-\alpha_k g_k. \tag{2.7}$$

Set $k=k+1$ and go to Step 1.

The difficulty in choosing α_k in Algorithm 2.1 is that we can not use the exact line search or the Armijo type line search.

For the exact line search, take the problem of minimizing the 1–norm of the variable in $\mathbb{R}^2$ for example, that is, to solve (1.1) with

$$f(x)=\|x\|_1,\quad x\in\mathbb{R}^2. \tag{2.8}$$

Suppose we let the initial vector be $x_1=(t_1\quad 0)^T$, where t_1 is a positive constant. For any positive constant t_2 in (0, 1), we can choose $g_1=(1\quad -t_2)^T$. Thus we have

$$x_2^T=[0\quad t_1t_2]. \tag{2.9}$$

Consequently, we may have

$$x_{2k+1}=\begin{bmatrix}\prod_{i=1}^{2k+1} t_i\\ 0\end{bmatrix},\quad x_{2k}=\begin{bmatrix}0\\ \prod_{i=1}^{2k} t_i\end{bmatrix}, \tag{2.10}$$

for all k, where t_i $(i=1, 2, \cdots)$ are any numbers in the interval $(0, 1)$. We may choose t_i such that $t^*=\prod_{i=1}^{\infty} t_i>0$. Then the only two accumulation points of $\{x_k\}$ are $(0, t^*)^T$ and $(t^*, 0)^T$. Neither of them is a stationary point of $\|x\|_1$.

Due to the nonsmoothness of $f(x)$, for any given constant $c_1\in(0, 1)$, the Armijo condition

$$f(x_k-\alpha\partial f(x_k))\leqslant f(x_k)-\alpha c_1\|\partial f(x_k)\|^2 \tag{2.11}$$

may fail for all $\alpha>0$.

Though both the exact line search and the Armijo type search may fail for the subgradient method, fortunately the angle between $-g_k$ and x^*-x_k is less than $\pi/2$, if x_k is not an optimal point of $f(x)$ and x^* is any point that solves (1.1). The following lemma is known; see, for example, Zowe (1985).

Lemma 2.2. *Assume $f(x)$ is convex and the set of optimal points of problem* (1.1)

$$X^*=\{x^*\,|\,f(x^*)=f^*=\min_{x\in R^n} f(x)\} \tag{2.12}$$

is nonempty. If x_k is not an optimal point and x^* is any point in X^*, there exists $T_k>0$ such that

$$\|x_k-\alpha g_k/\|g_k\|-x^*\|\leqslant\|x_k-x^*\| \tag{2.13}$$

for all $\alpha\in(0, T_k)$.

The subgradient method described in Algorithm 2.1 was developed and used by Shor (1962) to solve large scale transportation problems. Shor (1962) used the constant step, that is,

$$\alpha_k\equiv\alpha>0,\quad \text{for all } k. \tag{2.14}$$

Since $\|x_{k+1}-x_k\|=\alpha$ is bounded away from zero, it is easy to see that Algorithm 2.1 does not converge if α_k are defined by (2.14). However, one can show that

Theorem 2.3 (Shor, 1962). *If $f(x)$ is convex, and the set X^* is nonempty, for any given $\delta>0$, there exists a $\gamma>0$ such that for every given $\alpha\in(0, \gamma)$, Algorithm* 2.1 *ensures that*

$$\liminf_{k\to\infty} f(x_k)\leqslant f^*+\delta, \tag{2.15}$$

if α_k satisfy (2.14).

The above theorem shows that though the constant step method would not give convergence, it can solve problem (1.1) within any given accuracy by choosing small step–size.

The first technique of choosing α_k to force convergence for Algorithm 2.1 seems to be given by Ermol'ev (1966) and Poljak (1967) independently. They chose such α_k that satisfy

$$\alpha_k>0,\quad \lim_{k\to\infty}\alpha_k=0,\quad \sum_{k=1}^{\infty}\alpha_k=\infty. \tag{2.16}$$

The following theorem is due to Poljak (1967).

Theorem 2.4 (Poljak, 1967). *Assume $f(x)$ is convex, X^* is nonempty and bounded. Let x_k be generated by Algorithm 2.1 with (2.16). Then*

$$\lim_{k\to\infty} \rho(x_k, X^*)=0, \tag{2.17}$$

where the function $\rho(x, Y)$ has the value of the distance from the point x to the set Y, that is,

$$\rho(x, Y)=\min_{y\in Y}\|x-y\|. \tag{2.18}$$

Proof. Due to the convexity of $f(x)$, there exists a nondecreasing function $\delta(\varepsilon)>0$ such that for any positive ε the inequality

$$f(x)\leqslant f^*+\varepsilon \tag{2.19}$$

holds if

$$\rho(x, X^*)\leqslant\delta(\varepsilon). \tag{2.20}$$

For every k, we define

$$\varepsilon_k=f(x_k)-f^*. \tag{2.21}$$

Then if $\varepsilon_k>0$, due to the convexity of $f(x)$, we have

$$\begin{aligned}\|x_{k+1}-x^*\|^2&=\|x_k-x^*\|^2+\alpha_k^2-2\alpha_k(x_k-x^*)^T g_k/\|g_k\|\\&=\|x_k-x^*\|^2+\alpha_k^2-2\delta(\varepsilon_k)\alpha_k-2\alpha_k\left(x_k-x^*-\delta(\varepsilon_k)\frac{g_k}{\|g_k\|}\right)^T g_k/\|g_k\|\\&\leqslant\|x_k-x^*\|^2+\alpha_k^2-2\delta(\varepsilon_k)\alpha_k,\end{aligned} \tag{2.22}$$

which shows that

$$[\rho(x_{k+1}, X^*)]^2-[\rho(x_k, X^*)]^2\leqslant-\alpha_k(2\delta(\varepsilon_k)-\alpha_k). \tag{2.23}$$

The above inequality holds for all k if we define $\delta(0)=0$. Taking the sum over k from 1 to infinity on both sides of (2.23), we have

$$\liminf_{k\to\infty}\delta(\varepsilon_k)=0. \tag{2.24}$$

Consequently,

$$\liminf_{k\to\infty}\rho(x_k, X^*)=0. \tag{2.25}$$

If the theorem is not true, due to (2.25), there exist an $\varepsilon'>0$ and infinitely many k such that

$$\rho(x_{k+1}, X^*)>\rho(x_k, X^*) \tag{2.26}$$

and that

$$\varepsilon_k\geqslant\varepsilon'. \tag{2.27}$$

Now inequalities (2.23), (2.26) and (2.27) show that

$$2\delta(\varepsilon_k)<\alpha_k \tag{2.28}$$

for infinitely many k, which contradicts the facts that $\delta(\varepsilon_k)\geqslant\delta(\varepsilon')>0$ and that $\alpha_k\to 0$. Therefore the theorem is true. ∎

The above proof is a modification of that of Poljak (1967). Ermol'ev showed the same result under an additional condition that

$$\|g_k\|\leqslant C, \quad \text{for all } k, \tag{2.29}$$

for some positive constant C.

From the inequality

$$\|x_k - x^*\| + \|x_{k+1} - x^*\| \geqslant \alpha_k, \tag{2.30}$$

Algorithm 2.1 cannot converge faster than α_k tends to zero. Due to the condition $\sum_{k=1}^{\infty} \alpha_k = \infty$, Algorithm 2.1 with (2.16) never converges R-linearly. Here we say x_k converges to x^* R-linearly if there exist $q \in (0, 1)$ and $M > 0$ such that

$$\|x_k - x^*\| \leqslant M q^k \tag{2.31}$$

for all k (see Ortega and Rheinboldt, 1970).

To enforce R-rate convergence, Shor (1968) suggested that

$$\alpha_k = \alpha_0 q^k, \quad 0 < q < 1. \tag{2.32}$$

It is easy to see that $\sum_{k=i}^{\infty} \alpha_k < \infty$; hence the convergence result (2.17) is not valid. In fact, for any given α_0 and q, if the starting point x_1 is far away from the set of solution points $\left(\text{say, } \rho(x_1, X^*) > \frac{\alpha_0}{1-q}\right)$, the points $\{x_k\}$ will be bounded away from the set X^*. A positive result of the formula (2.32) is the following.

Theorem 2.5 (Shor, 1968). *Assume $f(x)$ is convex and there exists a positive constant l such that*

$$(x - x^*)^T \partial f(x) \geqslant l \|\partial f(x)\| \|x - x^*\| \tag{2.33}$$

for all x. Then there exist positive constants $\bar{q}$ and $\bar{\alpha}$ such that if $\bar{q} \leqslant q < 1$ and $\alpha_0 \geqslant \bar{\alpha}$, then Algorithm 2.1 with (2.32) will ensure that

$$\|x_k - x^*\| \geqslant M q^k \tag{2.34}$$

for some $x^* \in X^*$, where $\bar{q}$ and $\bar{\alpha}$ are dependent of l and $\|x_k - x^*\|$, and where M depends on the values q and α_0.

Though the rate of convergence is R-linear, (2.32) is not implementable in practice, since it requires information about the objective function $f(x)$ (for example, the value l and the distance $\|x_1 - x^*\|$). More details of the method are given by Goffin (1977).

The rate of convergence of Algorithm 2.1 can be improved if we know the optimal function value f^*. Such cases exist in practice, for example, in transferring a system of nonlinear inequalities into a minimax problem (see Poljak, 1978).

The following technique,

$$\alpha_k = \lambda \frac{(f(x_k) - f(x^*))}{\|g_k\|}, \quad 0 < \lambda < 2, \tag{2.35}$$

was first suggested by Eremin (1965) to solve the minimax problem

$$\min f(x) = \max_{1 \leqslant j \leqslant m} \{f_j(x), 0\}. \tag{2.36}$$

Poljak (1969) developed this technique in general nonsmooth optimization. The step-size rule (2.35), when applying to the minimax problem (2.36), coincides with the relaxation method for solving the inequality system

$$f_j(x) \leqslant 0, \quad 1 \leqslant j \leqslant m. \tag{2.37}$$

The relaxation method is traced back as far as Agmon (1954), and Motzkin and

Schoenberg (1954). For more details and its relation with nondifferentiable optimization, see Goffin (1978).

The following convergence theorem is given by Poljak (1969), which is also proved by Eremin (1965) for the minimax problem (2.36).

Theorem 2.6 (Poljak, 1969). *Assume $f(x)$ is convex, the set X^* is nonempty, and $f(x)$ satisfies*

$$\|\partial f(x)\| \leqslant C \tag{2.38}$$

for all x such that $\|x-x_1\| \leqslant \rho(x_1, X^*)$, and that

$$f(x)-f^* \geqslant l\rho(x, X^*) \tag{2.39}$$

for all x satisfying $\rho(x, X^*) \leqslant \rho(x_1, X^*)$. Then the sequence $\{x_k\}$ generated by Algorithm 2.1 with step-size rule (2.35) converges to a point $x^* \in X^*$ R-linearly, that is,

$$\|x_k-x^*\| \leqslant Mq^k, \tag{2.40}$$

where M is a constant, and q has the value

$$q=[1-\lambda(2-\lambda)l^2/C^2]^{\frac{1}{2}}<1. \tag{2.41}$$

Poljak (1969) also gives a modification of (2.35) for the case when the optimal function value f^* is unknown.

Shor (1970) introduced the space dilatation method to accelerate convergence of the subgradient method. The main idea of the method is to use not only g_k but also g_{k-1} to form the search direction at the k-th iteration. More details of the method can be found in Poljak (1978) and Zowe (1985). The following symmetrical form was suggested by Skokv (1974), and the notations are due to Zowe (1985).

Algorithm **2.7** (Space Dilatation Method).

Step 0: Given initial vector x_1, initial matrix $H_1=\alpha I$ for some positive α.

Step 1: Calculate a vector $g_k \in \partial f(x_k)$.

Step 2: Choose a step-size $\alpha_k>0$.

Step 3: Set

$$x_{k+1}=x_k-\alpha_k H_k g_k/(g_k^T H_k g_k)^{\frac{1}{2}}. \tag{2.42}$$

Step 4: Choose positive γ_k and β_k; set

$$H_{k+1}=\gamma_k\left(H_k-\beta_k \frac{H_k g_k g_k^T H_k}{g_k^T H_k g_k}\right). \tag{2.43}$$

Set $k=k+1$ and go to Step 1.

It is easy to see that H_k are all positive definite if $\beta_k<1$. One choice of α_k, β_k and γ_k was given by Shor (1977), where he chose constant parameters

$$\alpha_k=\frac{1}{n+1}, \quad \beta_k=\frac{2}{n+1}, \quad \gamma_k=\frac{n^2}{(n^2-1)} \tag{2.44}$$

for all k, where n is the dimension of the space. The following theorem is true; more details can be found in Goffin (1981), Shor (1977) and Zowe (1985).

Theorem 2.8. *Assume $f(x)$ is convex, the set X^* is nonempty and $\rho(x_1, X^*) \leqslant \gamma$. The sequence generated by Algorithm 2.7 with (2.44) satisfies*

$$\liminf_{k\to\infty} \frac{f(x_k)-f^*}{q^k}<\infty, \tag{2.45}$$

where q has the value

$$q=n[1-2/(n+1)]^{\frac{1}{2n}}/(n^2-1)^{\frac{1}{2}} \tag{2.46}$$

Many other space dilatation methods are discussed in Poljak (1978) and Shor (1983).

There are many extensions of the subgradient method, such as the finite difference approximation method of Gupal (1977) and the ellipsoid method of Shor (1977). For more details, see Lemarechal (1978a), Poljak (1978) and Zowe (1985).

§ 3. Cutting Plane Method

The cutting plane method for solving convex optimization problems was given by Cheney and Goldstein (1959) and Kelley (1959) independently. The method is constructed for finding the minimum value of a convex function $f(x)$ in a polyhedron S. At each iteration, a new hyperplane is introduced to reduce the polyhedron. All optimal points will remain in the new polyhedron, since no optimal points can lie in the region that is cut away from the original polyhedron. At the k-th iteration, the method requires solving the following linear programming problem

$$\min v, \quad v\in\mathbb{R}^1, \tag{3.1}$$

subject to $x\in S$ and

$$v\geqslant f(x_j)+g_j^T(x-x_j), \quad j=1, 2, \cdots, k. \tag{3.2}$$

Since $f(x)$ is convex, we have

$$f(x)=\sup_y \sup_{g\in\partial f(y)} [f(y)+g^T(x-y)]. \tag{3.3}$$

Hence the problem of minimizing $f(x)$ in the region S is equivalent to solving (3.1) subject to $x\in S$ and

$$v\geqslant f(y)+g^T(x-y) \tag{3.4}$$

for all $y\in S$ and $g\in\partial f(y)$. We can easily see that the linear system (3.2) is a finite approximation to the infinite system (3.4).

A formal description of the algorithm is as follows:

Algorithm **3.1** (The Cutting Plane Method).

Step 0: Given a polyhedron S, $x_1\in S$, set $k=1$.

Step 1: Compute $g_k\in\partial f(x_k)$.

Step 2: Solve (3.1)—(3.2) giving v_{k+1} and x_{k+1}.

Set $k=k+1$ and go to Step 1.

The convergence theorem of the above algorithm states that

Theorem 3.2 (Cheney and Goldstein, 1959). *Let $\{x_k\}$ and $\{v_k\}$ be generated by Algorithm* 3.1. *If $f(x)$ is convex and bounded below, then*

1) $v_2\leqslant v_3\leqslant\cdots\leqslant v_k\to f^*$;

2) *every accumulation point of $\{x_k\}$ is an optimal point of $f(x)$ in S.*

The method behaves very badly when $f(x)$ is smooth, since in this case $\nabla f(x_k)$ is very small for all large k, consequently (3.2) are similar systems for all large k. Another difficulty with the method is that the linear programming problem

(3.1)—(3.2) will have too many constraints and the simplex matrices tend to become singular for large k. To the author's knowledge, there has not been any result about the rate of convergence of the method. Hence the cutting plane method has never been in great favour, though it is the first method for solving general convex optimization problems. However, it is thought that a fast rate of convergence can be expected if the objective function $f(x)$ satisfies

$$\|f(x)-f(x^*)\|\geqslant m\|x-x^*\|, \quad \text{for all } x, \tag{3.5}$$

for some positive constant m.

§ 4. The Bundle Methods

The bundle methods are a class of descent methods for solving problem (1.1). The sequence $\{x_k\}$ generated by a bundle method has the property that $f(x_{k+1})\leqslant f(x_k)$ for all k.

A sub-class of bundle methods are conjugate subgradient methods, pioneered by Wolfe (1975). At the k-th iteration, a set $I_k\subset\{1, 2, \cdots, k\}$ is defined, and a search direction d_k is chosen so that $-d_k$ is the vector in the convex hull of $g_j\{j\in I_k\}$ which has the lease 2-norm. That is,

$$d_k=-\sum_{j\in I_k} w_j g_j, \tag{4.1}$$

where w_j $(j\in I_k)$ solve

$$\min\|\sum_{j\in I_k} w_j g_j\|^2 \tag{4.2}$$

subject to

$$\sum_{I_k} w_j=1, \quad w_j\geqslant 0. \tag{4.3}$$

A line search is made to obtain a step-size α_k. Define

$$y_{k+1}=x_k+\alpha_k d_k \tag{4.4}$$

and choose

$$g_{k+1}\in\partial F(y_{k+1}). \tag{4.5}$$

The line search technique requires that

$$g_{k+1}^T d_k\geqslant -m_1\|d_k\|^2, \tag{4.6}$$

where $m_1\in(0, 1)$ is a constant independent of k, and either

$$f(y_{k+1})\leqslant f(x_k)-m_2\alpha_k\|d_k\|^2 \tag{4.7}$$

(Armijo condition), or

$$\|y_{k+1}-x_k\|\leqslant m_3 s \tag{4.8}$$

(null step condition), where $m_2\in(0, m_1)$ and $m_3\in(0, 1)$ are constants, and $s>0$ is a small positive number (Lemarechal, 1980).

The following algorithm, stated in Lemarechal (1980), is a modification of Wolfe's (1975) original method.

Algorithm **4.1** (Conjugate Subgradient Method).

Step 0: Choose $x_1\in\mathbb{R}^n$, calculate $g_1\in\partial f(x_1)$.

Choose $0<m_2<m_2<\frac{1}{2}$, $0<m_3<1$, $s>0$ and $\eta>0$. Set $k=1$, $I_1=\{1\}$.

Step 1: Obtain d_k by solving (4.2)—(4.3). If $\|d_k\| \leqslant \eta$ then stop.

Step 2: Find y_{k+1} of the form (4.4) such that (4.5) holds and either (4.7) or (4.8) holds.

Step 3: Set $x_{k+1}=y_{k+1}$ if (4.6) holds, otherwise set $x_{k+1}=x_k$.

Step 4: Let $I_{k+1}=I_k \cup \{k+1\}T_k$, where $T_k \subset \{1, \cdots, k\}$ is the set of indices such that

$$\|y_l - x_{k+1}\| > \varepsilon. \tag{4.9}$$

Set $k=k+1$ and go to Step 1.

The following result is due to Wolfe (1975).

Theorem 4.2 (Wolfe, 1975). *Assume $f(x)$ is convex, and $\|\partial f(x)\|$ is bounded on an open set that contains the set $\{x|f(x) \leqslant f(x_1)\}$. Let $\{x_k\}$ be the sequence generated by Algorithm* 4.1. *If $f(x_k)$ is bounded below, then the algorithm stops after a finite number of iterations.*

If $f(x)$ is a quadratic convex function, if the line searches are exact and if we let $I_{k+1}=I_k \cup \{k\}$, then the directions are mutually conjugate, and consequently the algorithm finds an optimal point after at most n iterations.

The bundle methods are a generalization of conjugate subgradient methods. Instead of selecting a set I_k at every iteration, bundle methods introduce nonnegative weighted parameters $a_j^{(k)}$ $(j=1, 2, \cdots, k)$. At the k-th iteration, bundle methods compute $d_k=-\sum_{j=1}^{k} w_j g_j$, where w_j $(j=1, 2, \cdots, k)$ solve the problem

$$\min \|\sum_{j=1}^{k} w_j g_j\|^2, \tag{4.10}$$

subject to

$$\sum_{j=1}^{k} w_j = 1, \quad w_j \geqslant 0 \tag{4.11}$$

and

$$\sum_{j=1}^{k} w_j a_j^{(k)} \leqslant \varepsilon. \tag{4.12}$$

It is easy to see that (4.10)—(4.12) is equivalent to (4.2)—(4.3) if we let $a_j^{(k)}=0$ $(j \in I_k)$ and $a_j^{(k)}=\infty$ $(j \in I_k)$. The following implementation is due to Lemarechal (1980).

Algorithm **4.3** (Bundle Method).

Step 0: Choose $x_1 \in \mathbb{R}^n$, calculate $g_1 \in \partial f(x_1)$.

Choose $0<m_2<m_2<\frac{1}{2}$, $0<m_3<1$, $\varepsilon>0$ and $\eta>0$. Set $k=1$, $a_1^{(1)}=1$.

Step 1: Obtain d_k by solving (4.10)—(4.12). If $\|d_k\| \leqslant \eta$ then stop.

Step 2: Find y_{k+1} of the form (4.4) such that (4.5) holds and either (4.7) or

$$f(y_{k+1}) - \alpha_k g_{k+1}^T d_k \geqslant f(x_k) - \varepsilon \tag{4.13}$$

(null step condition) holds.

If (4.7) holds then go to Step 4.

Step 3: Set $x_{k+1}=y_{k+1}$, $a_{k+1}^{(k+1)}=1$, and

$$a_j^{(k+1)} = a_j^{(k)} + f(x_{k+1}) - f(x_k) - \alpha_k g_j^T d_k, \tag{4.14}$$

for $j=1, 2, \cdots, k$. Set $k=k+1$ and go to Step 1.

Step 4: Set $x_{k+1}=x_k$ and $a_j^{(k+1)}=a_j^{(k)}$ $(j=1, 2, \cdots, k)$.

Let

$$a_{k+1}^{(k+1)}=f(x_k)-f(y_{k+1})+\alpha_k g_{k+1}^T d_k. \tag{4.15}$$

Set $k=k+1$ and go to Step 1.

The convergence of the algorithm is stated as follows.

Theorem 4.4 (Lemarechal, 1978b). *Under the conditions of Theorem 4.2, the algorithm stops after a finite number of iterations.*

For other bundle methods, and the relation between bundle methods and other methods, such as the cutting plane method and the steepest descent method, see Lemarechal (1978a, 1978b, 1980), Lemarechal, Strodiot and Bihain (1981) and Zowe (1985).

§ 5. Trust Region Methods

Trust region methods are methods that replace the original problem by an easy-to-solve problem at every iteration. At each iteration a small region (normly a ball or box with x_k being the center) is chosen, and the approximate problem is solved within the region in order to obtain a trial step. The small region at each iteration is called a trust region, whose size is adjusted at every iteration. Normally, a trust region has the form $\{x \mid \|x-x_k\| \leqslant \Delta_k\}$, where Δ_k is called the trust region bound. Some tests are made to verify whether the trial step is "good", for example, to see whether the objective function can be reduced, that is, $f(x_k+d_k)<f(x_k)$. Basically, if the trial step is "good", then the trust region bound will not reduce and we accept the trial step (by which we mean that $x_{k+1}=x_k+d_k$); otherwise we reduce the trust region bound and resolve the approximation problem.

The trust region method was introduced by Levenberg (1944) to solve nonlinear least square problems. Levenberg's method was rediscovered by Marquardt (1963). For an overview of trust region methods for smooth optimization, see Fletcher (1980) and More (1982).

In this section, we consider trust region methods for solving the following composite problem

$$\min h(f(x)), \tag{5.1}$$

where $h(\cdot)$ is a convex function defined on $\mathbb{R}^m$ and is bounded below, and $f(x)=(f_1(x), f_2(x), \cdots, f_m(x))^T$ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ and $f_j(x)(j=1, 2, \cdots, m)$ are all continuously differentiable functions on $\mathbb{R}^n$.

At the k-th iteration, we let the approximation problem is

$$\min \phi_k(d) \equiv h(f(x_k)+(\nabla f(x_k))^T d)+\frac{1}{2} d^T B_k d, \quad d \subset \mathbb{R}^n, \tag{5.2}$$

subject to

$$\|d\| \leqslant \Delta_k, \tag{5.3}$$

where $\Delta_k>0$ is the trust region bound at the k-th iteration. Let d_k be a solution of (5.2)—(5.3). Then the ratio

$$r_k=\frac{h(x_k)-h(x_k+d_k)}{h(x_k)-\phi_k(d_k)} \tag{5.4}$$

is calculated, which plays an important role in adjusting the trust region bound for the next iteration and in choosing the next *iterate* x_{k+1}.

Fletcher (1982a) first considered trust region methods for solving the composite problem (5.1). The following algorithm is a modification of Fletcher's (1982a) model algorithm.

Algorithm **5.1** (Model Trust Region Algorithm).

Step 0: Given $x_1 \in \mathbb{R}^n$, $B_1 \in \mathbb{R}^{n\times n}$.
Choose $c_j > 0$ $(j=1, 2, 3, 4)$ that $c_2 < 1 \leqslant c_1$ and $c_3 \leqslant c_4 < 1$.
Set $k=1$.

Step 1: Solving (5.2)—(5.3), giving d_k.
Calculate r_k by (5.4).

Step 2: Set

$$x_{k+1}=\begin{cases} x_k+d_k, & \text{if } r_k>0; \\ x_k, & \text{otherwise,} \end{cases} \tag{5.5}$$

and let Δ_k satisfy

$$\Delta_{k+1}\subset\begin{cases} [\|d_k\|, \min(c_1\Delta_k, \bar{\Delta})], & \text{if } r_k \geqslant c_2; \\ [c_3\|d_k\|, c_4\Delta_k], & \text{otherwise.} \end{cases} \tag{5.6}$$

Step 3: Update B_k, set $k=k+1$ and go to Step 1.

The following convergence result was given by Fletcher (1981).

Theorem 5.2 (Fletcher, 1981). *Let* $\{x_k\}$ *be generated by Algorithm* 5.1 *with* B_k *uniformly bounded. Then* $\{x_k\}$ *is not bounded away from stationary points of problem* (5.1).

The above theorem is still true if we replace the boundedness of B_k by some conditions that can be normally satisfied by a certain updating formula and that cannot imply the boundedness explicitly. One can prove that the following result is valid:

Theorem 5.3 (Yuan, 1985a). *Theorem* 5.2 *is still true if we replace the boundedness of* $\{B_k\}$ *by either*

$$\|B_k\| \leqslant c_5 + c_6 \sum_{j=1}^{k} \Delta_j \tag{5.7}$$

for all k, *or by*

$$\|B_k\| \leqslant c_7 + c_8 k \tag{5.8}$$

for all k, *where* c_j $(j=5, 6, 7, 8)$ *are any given positive constants.*

The rate of convergence of Algorithm 5.1 has been studied by many authors, including Powell and Yuan (1984) and Womersley (1985).

Assumption **5.4**.

1) $x_k \to x^*$,

2) The second order sufficiency condition holds, that is,

$$d^T W^* d > 0 \tag{5.9}$$

for all d satisfying $P^* d = 0$, where P^* is a projector from $\mathbb{R}^n$ to the null space of $(A(x^*))^T$ and where W^* is the matrix $\sum_{j=1}^{m} \lambda_j^* \nabla^2 f(x^*)$ and $\lambda_j^* (j=1, \cdots, m)$ satisfy $\sum_{j=1}^{m} \lambda_j^* \nabla f(x^*) = 0$.

Under certain conditions, including Assumption 5.4 one can prove

Theorem 5.5 (Powell and Yuan, 1984, Womersley, 1985). *x_k converges to x^* Q-superlinearly (Ortega and Rheinboldt, 1970), if and only if*

$$\lim_{k\to\infty}\frac{\|P^*(W^*-B_k)d_k\|}{\|d_k\|}=0, \tag{5.10}$$

and the trust region bound is inactive for all large k.

Unfortunately, it is not always true for the inactive trust region bound. Actually, by constructing a minimax problem, Yuan (1984) showed that it is not possible to prove a superlinear convergence result for Algorithm 5.1.

One way of improving the rate of convergence is due to Fletcher (1982b). A second order correction method proposed by Fletcher (1982b) is stated below:

Algorithm **5.6** (Fletcher, 1982b).

Step 0: Given $x_1\in\mathbb{R}^n$, $B_1\in\mathbb{R}^{n\times n}$.
Choose $c_j>0$ $(j=1, 2, 3, 4)$ that $c_2<1\leqslant c_1$ and $c_3\leqslant c_4<1$.
Set $k=1$.

Step 1: Solving (5.2)—(5.3), giving d_k.
Calculate r_k by (5.4). If $r_k>0.75$ then go to Step 7.

Step 2: Solve

$$\min \psi_k(d)=\phi_k(d_k+d)+h(f(x_k+d_k)+A_k^Td_k) -h(f(x_k)+A_k^T[d_k+d]), \tag{5.11}$$

subject to

$$\|d_k+d\|\leqslant\Delta_k. \tag{5.12}$$

Let $\bar{d}_k$ be a solution of (5.11)—(5.12).
Define

$$r_e^{(k)}=r_k+\frac{\psi_k(0)-\psi_k(\bar{d}_k)}{\phi_k(0)-\phi_k(d_k)}. \tag{5.13}$$

If $r_k<0.25$ then go to Step 4.

Step 3: If $r_e^{(k)}\in[0.9, 1.1]$, set $\Delta_{k+1}=2\Delta_k$ and go to Step 9.
Otherwise go to Step 8.

Step 4: If $r_e^{(k)}\in[0.75, 1.25]$ go to Step 5.
Evaluate the ratio

$$\bar{r}_k=\frac{h(f(x_k))-h(f(x_k+d_k+\bar{d}_k))}{\phi_k(0)-\phi_k(d_k)}. \tag{5.14}$$

Assign $d_k:=d_k+\bar{d}_k$, $r_k:=\bar{r}_k$.
If $r_k>7.5$ go to Step 7.
If $r_k\geqslant 0.25$ go to Step 8.

Step 5: Set $\Delta_{k+1}=\alpha_k\|d_k\|$, $\alpha_k\in[0.1, 0.5]$.
If $r_k>0$ then go to Step 9.

Step 6: $x_{k+1}=x_k$, go to Step 10.

Step 7: If $\|d_k\|<\Delta_k$ then go to Step 8.
If $r_k>0.9$ then $\Delta_{k+1}:=4\Delta_k$ else $\Delta_{k+1}=2\Delta_k$.
Go to Step 9.

Step 8: $\Delta_{k+1}=\Delta_k$.

Step 9: Set $x_{k+1}:=x_k+d_k$.

Step 10: Generate B_{k+1}, set $k=k+1$ and go to Step 1.

The convergence of the method can be proved similarly to Theorem 5.2. For more details, see Fletcher (1982b).

The superlinear convergence of Algorithm 5.6 was proved by Yuan (1985b). We state the result as follows.

Theorem 3.7 (Yuan, 1985b). *If the function $h(f)$ is a polyhedral convex function, if the gradients $f_j(x^*)(j=1, 2, \cdots, m)$ are linearly independent, if the first and second order sufficiency condition holds, and if $x_k\to x^*$ and $\lambda_k\to\lambda(x^*)$, then Algorithm 5.6 converges to x^* Q-superlinearly, if B_k tends to the Hessian of the Lagrangian at the solution.*

We believe the above theorem holds for any general convex function, but we have not produced a sound proof for our conjecture.

Trust region methods for composite optimization problems are also studied by many other authors, including Burke (1985), where convergence properties of a class of methods are analyzed.

An obvious difficulty for the methods discussed in this section is how to solve the sub-problem (5.2)—(5.3). Since the objective function in (5.2) is the sum of a convex function and a quadratic function, we still need to apply some other methods for nonsmooth optimization to solve (5.2)—(5.3), if $h(\cdot)$ is a general convex function. However, if $h(\cdot)$ is a polyhedral convex function, we can rewrite (5.2) into a linearly constrained quadratic optimization problem.

§ 6. Acknowledgement

I wish to thank my supervisor, Professor M. J. D. Powell, for his constant help and encouragement.

References

[1] S. Agmon, The relaxation method for linear inequalities, *Canadian J. Math.*, **6** (1954), 382—392.

[2] M. L. Balinski, P. Wolfe (eds.), Nondifferentiable Optimization, *Mathematical Programming Studies*, **3** (1975), North-Holland, Amsterdam.

[3] J. V. Burke, Descent methods for composite nondifferentiable optimization problems, *Mathematical Programming*, **33** (1985), 260—279.

[4] E. W. Cheney, A. A. Goldstein, Newton's method for convex programming and Chebyshev approximation, *Numerische Mathematik*, **1** (1959), 253—268.

[5] F. H. Clarke, Generalized gradients and applications, *Trans. of Amer. Math. Soc.*, **205** (1975), 247—262.

[6] I. I. Eremin, A Generalization of the Motzkin-Agmon relaxation method, *Soviet Math. Doklady*, **6** (1965), 219—221.

[7] Yu. M. Ermol'ev, Method of solution of nonlinear extremal problems, *Kibernetika*. **2**: 4 (1966), 1—17. (in Russian)

[8] R. Fletcher, Practical Methods of Optimization, Vol. 1: Unconstrained Optimization. John Wiley & Sons, New York, 1980.

[9] R. Fletcher, Practical Methods of Optimization, Vol. 2: Constrained Optimization, John Wiley & Sons, New York, 1981.

[10] R. Fletcher, A model algorithm for composite NDO problem, *Mathematical Programming Studies*, **17** (1982), 67—76. (1982a)

[11] R. Fletcher, Second order corrections for nondifferentiable optimization, in: G. A. Watson, ed.,

Numerical Analysis, Springer–Verlag, Berlin, 1982, 85—114. (1982b)

[12] J. L. Goffin, On the convergence rates of subgradient optimization methods, *Mathematical Programming*, **13** (1977), 329—344.

[13] J. L. Goffin, Nondifferentiable optimization and the relaxation method, in: C. Lemarechal, R. Miffin, eds., *Nonsmooth Optimization*, Pergamon Press, Oxford, 1978, 31—49.

[14] J. L. Goffin, Convergence results in a class of variable metric subgradient methods, in: O. L Mangasarian, R. R. Meyer, S. M. Robinson, eds., *Nonlinear Programming*, **4**, Academic Press, New York, 1981.

[15] A. M. Gupal, On a minimization method for almost–differentiable functions, *Kibernetika*, **13**: 1 (1977), 114—116. (in Russian)

[16] J. E. Kelley, The cutting plane method for solving convex programs, *J. of SIAM*, **8** (1960), 703—712.

[17] C. Lemarechal, Bundle methods in nonsmooth optimization, in: C. Lemarechal, R. Mifflin, eds., *Nonsmooth Optimization*, Pergamon Press, Oxford, 1978, 79—102. (1978a)

[18] C. Lemarechal, Nonsmooth optimization and descent methods, RR–78–4, IIASA Report, 1978. (1978b)

[19] C. Lemarechal, Nondifferentiable optimization, in: L. C. W. Dixon, E. Spedicato, G. P. Szego, eds., *Nonlinear Optimization*, Birkhauser, Boston, 1980, 149—199.

[20] C. Lemarechal, R. Mifflin (ed.), *Nonsmooth Optimization*, IIASA Proceeding 3 (Pergamon, Oxford, 1978).

[21] C. Lemarechal, J. J. Strodiot, A. Bihain, On a bundle algorithm for nonsmooth optimization, in: O. L. Mangasarian, R. R. Meyer, S. M. Robinson, eds., *Nonlinear Programming*, **4**, Academic Press, New York, 1981.

[22] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.*, **2** (1944), 164—166.

[23] D. W. Marquardt, An algorithm for least–squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, **11** (1963), 431—441.

[24] J. J. More, Recent developments in algorithms and software for trust region methods, *Report ANL/MCS–TM2*, Argonne National Laboratory, Illinois, 1982.

[25] T. Motzkin, I. J. Schoenberg, The relaxation method for linear inequalities, *Canadian J. Math.*, **6** (1954), 393—404.

[26] E. A. Nurminskii, Progress on Nondifferentiable Optimization, IIASA Proceeding, Laxenburg, 1982.

[27] J. M. Ortega, W. C. Rheinboldt, Iterative Solution of Non–linear Equations in Several Variables, Academic Press, New York, 1970.

[28] B. T. Poljak, A general method of solving extremal problems, *Soviet Math. Doklady*, **8** (1967), 593—597.

[29] B. T. Poljak, Minimization of unsmooth functionals, *USSR Computational Math. and Math. Physics*, **9** (1969), 14—29.

[30] B. T. Poljak, Subgradient methods: A Survey of Soviet Research, in: C. Lemarechal, R. Mifflin, eds., *Nonsmooth Optimization*, Pergamon Press, Oxford, 1978.

[31] M. J. D. Powell, Y. Yuan, Conditions for superlinear convergence in I_1 and I_2 solutions of overdetermined nonlinear equations, *IMA J. Numer. Anal.*, **4** (1984), 241—251.

[32] N. Z. Shor, Application of the gradient method for the solution of network transportation problems, Notes, Scientific Seminar on Theory and Application of Cybernetics and Operations Research, Academy of Sciences, Kiev, 1962. (in Russian)

[33] N. Z. Shor, The rate of convergence of the generalized gradient descent method, *Kibernetika*, **4**: 3 (1968), 98—99. (in Russian)

[34] N. Z. Shor, Utilization of the operation of space dilatation in the minimization of convex functions, *Kibernetika*, **6**: 1 (1970), 6—12. (in Russian)

[35] N. Z. Shor, Cut–off method with space extention in convex programming problems, *Kibernetika*, **13**: 1 (1977), 94—95. (in Russian)

[36] N. Z. Shor, Generalized gradient methods of nondifferentiable optimization emplying space dilatation operations, in: A. Bachem, M. Grotschel, B. Korte, eds., Mathematical Programming: The State of the Art, Springer–Verlag, Berlin, 1983, 501—529.

[37] V. A. Skokv, Note on minimization methods using operation of space dilatation, *Kibernetika*, **10**: 4 (1974), 115—117. (in Russian)

[38] R. S. Womersley, Local properties of algorithms for minimizing nonsmooth composite functions, *Mathematical Programming*, **32** (1985), 69—89.

[39] P. Wolfe, A method of conjugate subgradients for minimizing nondifferentiable functions, *Mathematical Programming Study*, **3** (1975), 145—173.

[40] Y. Yuan, An example of only linear convergence of trust region algorithms for nonsmooth optimization, *IMA J. Numer. Anal.*, **4** (1984), 327—335.

[41] Y. Yuan, Conditions for convergence of trust region algorithms for nonsmooth optimization, *Mathematical Programming*, **31** (1985), 220—228. (1985a)

[42] Y. Yuan, On the superlinear convergence of a trust region algorithm for nonsmooth optimization, *Mathematical Programming*, **31** (1985), 269—285. (1985b)

[43] J. Zowe, Nondifferentiable optimization, in: K. Schittkowski, ed., Computational Mathematical Programming, Springer-Verlag, Berlin, 1985, 323—356.