NOTES ON NONSMOOTH OPTIMIZATION

Masao FUKUSHIMA (Kyoto University)

福島雅夫 (京都大学)

1. Introduction

Non-smooth Optimization (NSO) or Nondifferentiable Optimization (NDO) deals with optimization problems whose objective and constraint functions are not necessarily differentiable. The non-smooth functions which are encountered in practice are often defined as the max function

\[ f(x) = \max \{ F(x,y) \mid y \in Y \}, \tag{1.1} \]

where \( F \) is assumed to be smooth with respect to \( x \). Some examples of non-smooth functions are as follows.


Let \( m \) facilities be situated at points \( a_1, a_2, \ldots, a_m \) on a plane. Suppose that \( n \) new facilities should be located and that cost should be imposed on transportation of goods between the new facilities and the existing facilities and between the new facilities themselves. Then the problem of efficiently locating the new facilities may be formulated as

\[ \text{minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji}(\|x_j-a_i\|) + \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} g_{jk}(\|x_j-x_k\|), \tag{1.2} \]

where \( x_1, x_2, \ldots, x_n \) are the locations of the new facilities to be found, the functions \( f_{ji} \) and \( g_{jk} \) are nondecreasing, and \( \|\cdot\| \) is a norm in \( \mathbb{R}^2 \). Since \( \|\cdot\| \) has discontinuous first derivatives at the origin, the objective function of problem (1.2) is usually nonsmooth.

Example 2. Exact penalty function: Fletcher (1981).

Consider the nonlinear programming problem

\[ \text{minimize} \quad f(x) \tag{1.3} \]

subject to \( c_i(x) = 0, \quad i=1,2,\ldots,m' \)

\[ c_i(x) \leq 0, \quad i=m'+1,\ldots,m. \]
where \( f \) and \( c_i \) are assumed to be smooth. Then it is well known that, under some regularity conditions, a solution of problem (1.3) may be obtained by solving the unconstrained problem

\[
\text{minimize } F_r(x),
\]

where \( F_r \) is defined by

\[
F_r(x) = f(x) + r\left[ \sum_{i=1}^{m} |c_i(x)| + \sum_{i=m+1}^{m'} \max(0, c_i(x)) \right]
\]

and \( r \) is a sufficiently large positive constant. The function \( F_r \) is clearly nonsmooth.

**Example 3.** Decomposition of large problems: Fukushima (1987).

Consider the nonlinear programming problem

\[
\begin{align*}
\text{minimize} & \quad F(x) + \langle q, y \rangle \\
\text{subject to} & \quad Ax + By \leq b, \\
& \quad c_i(x) = 0, \quad i=1,2,...,m,
\end{align*}
\]

where \( F \) and \( c_i \) are smooth, \( A, B, q \) and \( b \) are matrices or vectors of appropriate dimension, and \( \langle \cdot, \cdot \rangle \) denotes the inner product. Then, as in Benders' decomposition method, we may rewrite (1.5) as

\[
\begin{align*}
\text{minimize} & \quad F(x) + f(x) \\
\text{subject to} & \quad c_i(x) = 0, \quad i=1,2,...,m,
\end{align*}
\]

where \( f \) is given by

\[
f(x) = \min \{ \langle q, y \rangle \mid By \leq b - Ax \} = \max \{ \langle Ax-b, w \rangle \mid B^Tw = -q, \ w \geq 0 \}.
\]

(The last equality follows from the duality theory in linear programming.) The function \( f \) is a (polyhedral) convex function which is generally nonsmooth.


The equilibrium traffic assignment problem and the optimal routing problem in a packet switched communication network can be formulated as a convex cost multicommodity flow problem. Consider a directed network with the set \( N \) of nodes and the set \( A \) of arcs. Let \( K \) denote the
set of commodities to be transported through the network. Assume that
each commodity \( k \in K \) has a single origin-destination (OD) pair \((s_k, t_k)\)
and let \( P_k \) be the set of paths between the OD pair \((s_k, t_k)\). Then the
minimum cost multicommodity flow problem may be stated as

\[
\text{minimize } \sum_{a \in A} f_a\left( \sum_{k \in K} \sum_{p \in P_k} \delta_{ap} y_p \right) \\
\text{subject to } \sum_{p \in P_k} y_p = D_k, \quad \forall k \in K, \\
y_p \geq 0, \quad \forall p \in P_k, \quad \forall k \in K,
\]

where \( f_a \) is the cost function for arc \( a \), \( x_a \) is the total flow on
arc \( a \), \( y_p \) is the flow (of commodity \( k \)) along path \( p \), \( D_k \) is the
given nonnegative flow requirement for commodity \( k \), and \( \delta_{ap} \) are
elements of the arc-path incidence matrix associated with the given
network. If we assume that the functions \( f_a \) are convex, then the dual
of problem (1.8) may be given as

\[
\text{minimize } \sum_{a \in A} f_a^*(u_a) + \sum_{k \in K} D_k h_k(u),
\]

where \( f_a^* \), the conjugate function of \( f_a \), is defined by

\[
f_a^*(u_a) = \sup \{ x_a u_a - f_a(x_a) \mid x_a \in \mathbb{R} \},
\]

and \( h_k \) is defined by

\[
h_k(u) = -\min\{ \sum_{a \in A} u_a \delta_{ap} \mid p \in P_k \}.
\]

Note that both \( f_a^* \) and \( h_k \) are convex. In particular, since \( f_a^* \) is a
function of a single variable, it is often possible to obtain an explic-
it representation of \( f_a^* \) from (1.10). Moreover, (1.11) indicates that,
for each \( k \), \( h_k(u) \) can be calculated by finding a shortest path between
the OD pair \((s_k, t_k)\) in the network with arc lengths given by \( u_a, a \in A \).

2. Algorithms

The key tool of dealing with nonsmooth functions is the notion of
subgradient (Rockafellar, 1970). Let \( f \) be a convex function on \( \mathbb{R}^n \).
Then, a vector \( g \) is called a subgradient of \( f \) at \( x \) if

\[
f(x') - f(x) \geq <g, x' - x> \quad \text{for all } x'.
\]
The set of subgradients of \( f \) at \( x \) is denoted by \( \partial f(x) \). The notion of subgradient has been generalized to various classes of nonconvex functions. (See, e.g., Clarke (1983).)

Roughly speaking, we may classify two different types of nonsmooth functions, depending upon the information available. Womersley and Fletcher (1986) use the terminology, "composite nonsmooth problems" and "basic nonsmooth problems". For the former, there is sufficient information available at each \( x \) to calculate the set \( \partial f(x) \) of subgradients completely, while for the latter, the set of subgradients must be approximated using information evaluated at various points around \( x \). When the nonsmooth function is defined as a max function of the form (1.1), the former correspond to the case where the set \( Y \) is finite and all component functions \( F(\cdot, y) \) are easily enumerated, and the latter correspond to the case where the number of elements in \( Y \) is either infinite or too many to enumerate individually. Examples 1 and 2 in the previous section belong to the class of composite nonsmooth problems, and Examples 3 and 4 typify the basic nonsmooth problems.

2.1. Algorithms for Composite Nonsmooth Problems

Typical composite nonsmooth problems are represented as

\[
\text{minimize } \phi(x) \overset{\triangle}{=} F(x) + h(c(x)),
\]

where \( F: \mathbb{R}^n \to \mathbb{R} \) and \( c: \mathbb{R}^n \to \mathbb{R}^m \) are smooth and \( h: \mathbb{R}^m \to \mathbb{R} \) is a polyhedral convex function. Fletcher (1981) presents an algorithm which constructs a quadratic approximation of problem (2.2) at each trial point and uses a trust region technique to guarantee the validity of the quadratic model as a good approximation to the original problem. Specifically, the algorithm of Fletcher (1981) solves at each iteration the following subproblem:

\[
\text{minimize } q^k(d) + h(g^k(d))
\]

subject to \( \|d\|_2 \leq \Delta^k \)

where
\[ q^k(d) = F(x^k) + \nabla F(x^k)^T d + \frac{1}{2} d^T W^k d, \]
\[ \ell^k(d) = c(x^k) + \nabla c(x^k)^T d, \]
\[ W^k = \nabla^2 F(x^k) + \sum_{i=1}^{M} \lambda_i^k \nabla^2 c_i(x^k), \]
\[ \lambda_i^k \] are estimates of the optimal Lagrange multipliers, and \( \Delta^k \) is the radius of the trust region. Note that problem (2.3) can be transformed into a quadratic programming problem.

Let \( d^k \) be a solution of (2.3). If \( d^k \) gives a sufficient reduction in the objective function \( \phi \) of the original problem (2.2), then the next iterate \( x^{k+1} \) is chosen to be \( x^k + d^k \). Otherwise, the radius of the trust region has to be controlled to enforce the reduction of the function \( \phi \). Fletcher (1981) shows that, under suitable conditions, such a step restriction strategy generates a sequence whose accumulation point satisfies the optimality conditions for (2.2).

Concerning the rate of convergence, it is known that the trust region algorithm of Fletcher (1981) suffers from the so-called Maratos effect which may hinder superlinear convergence. To overcome this difficulty, Fletcher (1982) proposes to solve another subproblem of the following form, after the subproblem (2.3) is solved:

\[ \text{minimize } q^k(d) + h(\bar{c}^k + \nabla c(x^k)^T d) \tag{2.4} \]
\[ \text{subject to } \|d\| \leq \Delta^k, \]
where \( \bar{c}^k = c(x^k + d^k) - \nabla c(x^k)^T d^k \) and \( d^k \) is the solution of (2.3). Fletcher (1981) establishes a superlinear convergence result for this modified algorithm.

More recently, Yamakawa, Fukushima and Ibaraki (1989) have proposed another remedy of overcoming the Maratos effect, using the idea given by Fukushima (1986b) for the SQP methods in nonlinear programming. Specifically, this algorithm solves, instead of (2.4), the following subproblem:

\[ \text{minimize } \hat{q}^k(d) + h(\hat{c}^k(d)) \tag{2.5} \]
\[ \text{subject to } \|d\| \leq \Delta^k, \]
where
\[ \hat{\alpha}^k(d) = F(x^k) + (p^k)^T d + \frac{1}{2} d^T W^k d, \]
\[ \hat{\lambda}^k_i(d) = c_i(x^k) + (a^k_i)^T d, \]
\[ p^k = \nabla F(x^k) - \sum_{i=1}^{m} \lambda^k_i r^k_i, \]
\[ a^k_i = \nabla c_i(x^k) + r^k_i, \quad i=1,2,...,m, \]
\[ r^k_i = \frac{1}{2} \nabla^2 c_i(x^k) d^k. \]

Note that the calculation of the above vectors requires no extra work of evaluating the second derivatives, because they have already been used to evaluate the matrix \( W^k \) in subproblem (2.3). In Yamakawa, Fukushima and Ibaraki (1989), it is shown that the Maratos effect can be avoided by using the solutions of subproblems (2.5) and very promising numerical results are reported for some ill-conditioned test problems.

### 2.2 Algorithms for Basic Nonsmooth Problems

For a function involved in basic nonsmooth problems, it is usually assumed that only one subgradient is available at each point. Early numerical methods that can deal with nonsmooth functions are subgradient methods and cutting plane methods. In particular, the subgradient method is one of the antecedents of the ellipsoid method for linear programming. The subgradient method and the ellipsoid method are fully described by Shor (1985) and Bland, Goldfarb and Todd (1981), respectively. The cutting plane method, which was independently developed by Cheney and Goldstein and Kelley about thirty years ago, has played an important role in nonsmooth optimization. The cutting plane method may be viewed as a predecessor of various descent methods developed by a number of authors including Lemarechal, Wolfe, Mifflin, Fukushima, Kiwiel and Auslender (see, e.g., Kiwiel (1985) and the references therein.) The primitive cutting plane method has the drawback of accumulating cutting planes infinitely. An attempt to overcome this difficulty is proposed by Fukushima (1983) for nonsmooth convex programs and this idea has been extended further to variational inequalities by Fukushima (1986a).
In the rest of this section, we briefly describe a class of descent methods which may be considered a modification of the proximal method studied by Rockafellar (1976).

First let us consider the unconstrained problem
\[ \minimize \ f(x), \]  
where \( f \) is a convex function. Given the current iterate \( x \), the proximal method (approximately) solves the subproblem
\[ \minimize \ \frac{\lambda}{2} \| p \|^2 + f(x+p), \]  
where \( \lambda \) is a positive parameter. Once the solution \( p \) of (2.7) is obtained, the next iterate \( x^+ \) is then determined by \( x^+ = x + p \).

Fukushima (1984a) observes that if \( \bar{p} \) solves (2.7), then the vector \( \bar{g} = \frac{1}{\lambda} \bar{p} \) is the minimum norm element of \( \partial_{\epsilon} f(x) \), the set of \( \epsilon \)-subgradients of \( f \) at \( x \), defined by
\[ \partial_{\epsilon} f(x) = \{ g \mid f(x') - f(x) \geq \langle g, x' - x \rangle - \epsilon, \forall x' \}, \]  
where \( \epsilon \geq 0 \). Thus \( \bar{p} \) is a descent direction of \( f \) at \( x \), and the next iterate \( x^+ \) may be determined as
\[ x^+ = x + \alpha \bar{p}, \]  
where the step-size \( \alpha > 0 \) is chosen in such a way that a sufficient reduction in the objective value is obtained. Another important fact is that solving (2.7) only approximately suffices to guarantee convergence of the whole algorithm. To solve (2.7), Fukushima (1984a) uses a cutting plane technique in which cutting planes are generated only for the convex term \( f(x+p) \) and hence search direction \( p \) is obtained by solving a sequence of problems of the form
\[ \minimize \ \frac{\lambda}{2} \| p \|^2 + \hat{f}(p), \]  
where \( \hat{f} \) is a polyhedral (outer) approximation of \( f \). Note that (2.10) can easily be transformed into a quadratic programming problem. Recently, Auslender (1987) extended this idea to get a more general method.

The proximal method has been generalized to the nonconvex problem
\[ \minimize \ F(x) + f(x), \]  
where \( F \) is smooth but not necessarily convex, and \( f \) is convex but
not necessarily smooth. Fukushima and Mine (1981) propose to determine a search direction at \( x \) by solving the "convex" subproblem

\[
\text{minimize} \quad \frac{\lambda}{2} \| p \|^2 + \langle \nabla F(x), p \rangle + f(x+p),
\]

(2.12)

where \( \lambda \) is a positive parameter. It can be shown that the solution of (2.12) is a descent direction of the objective function of (2.11), so that the next iterate may be found by line search. The algorithm presented by Fukushima and Mine (1981) is conceptual in the sense that it requires the exact solution of (2.12) which is usually impractical. Recently Kiwiel (1986) has improved it to obtain an implementable algorithm.

The proximal method can further be extended to a class of constrained nonsmooth nonconvex optimization problems (Fukushima, 1987). Let us consider the problem

\[
\text{minimize} \quad F(x) + f(x)
\]

subject to \( c_i(x) = 0, \ i = 1, 2, \ldots, m, \)

where \( F \) and \( f \) are the same as in (2.11), and \( c_i \) are assumed to be smooth. We note here that the algorithm to be described can deal with smooth inequality constraints by slight modification. Given the current iterate \( x \), the direction finding subproblem can be defined by

\[
\text{minimize} \quad \frac{1}{2} \langle Bp, p \rangle + \langle \nabla F(x), p \rangle + f(x+p)
\]

subject to \( c(x) + A(x)p = 0, \)

where \( B \) is symmetric and positive definite, \( c(x) = (c_1(x), \ldots, c_m(x))^T \) and \( A(x) = \nabla c(x)^T \). Note that the matrix \( \lambda I \) used in (2.7) and (2.12) has been replaced by a more general matrix \( B \) in (2.14). Problem (2.14) is a linearly constrained "convex" programming problem. Again we can apply the cutting plane technique to find an approximate solution of (2.14). That is, we solve a sequence of problems of the form

\[
\text{minimize} \quad \frac{1}{2} \langle Bp, p \rangle + \langle \nabla F(x), p \rangle + \hat{f}(p)
\]

subject to \( c(x) + A(x)p = 0, \)

where \( \hat{f} \) is a polyhedral (outer) approximation of \( f \), defined by the cutting planes generated so far. Problem (2.15) can be transformed into
a convex quadratic programming problem. Once an approximate solution of (2.14) satisfying some criteria is found, then the next iterate \( x_+ \) is determined by line search using the exact penalty function

\[
F_r(x) = F(x) + f(x) + r \sum_{i=1}^{m} |c_i(x)|.
\]

(2.16)

The penalty parameter \( r > 0 \) must be large enough to guarantee global convergence to a solution of (2.13), but we do not know a priori how large it should be. Therefore we have to adjust \( r \) to a suitable value automatically in the course of solving QP subproblems (2.15). This can be done using information on the optimal Lagrange multipliers of QP subproblems (2.15). We can show that, under some standard assumptions, the penalty parameter remains constant after a finite number of iterations and, moreover, an approximate optimal solution of any desirable accuracy can be obtained for the original problem (2.13) after finitely many steps.

To summarize, the proximal method can be extended in conjunction with descent methods to solve various nonsmooth optimization problems. In particular, the method presented by Fukushima (1987) may be viewed as a natural generalization of the globally convergent successive quadratic programming (SQP) methods for smooth nonlinear programming problems.

References


