G.L. Nemhauser et al., Eds., *Handbooks in OR & MS, Vol. 1*
© Elsevier Science Publishers B.V. (North-Holland) 1989

Chapter VII

Nondifferentiable Optimization

Claude Lemaréchal

INRIA, Domaine de Voluceau, BP 105 – Rocquencourt, 78153 Le Chesnay, France

There are many situations in operations research where one has to optimize a function which fails to have derivatives for some values of the variables. This is what Nondifferentiable Optimization (NDO) or Nonsmooth Optimization (NSO) deals with. For this kind of situation, new tools are required to replace standard differential calculus, and these new tools come from convex analysis.

Section 1 contains the necessary concepts and the essential basic properties, while some examples of practical problems motivating the use of NSO are listed in Section 2. In Section 3, we show how and why classical methods fail. Section 4 is devoted to some possibilities that can be used when a special structure exists in the nonsmooth problem. The so-called subgradient methods, coming mainly from the Soviet Union (Kiev), are studied in Section 5, and more recent methods, mainly developed in the West (the bundle methods), in Section 6. Finally, we give in Section 7 some orientations for future research; the relevant literature is reviewed in Section 8 with its bibliography.

Our development is by no means original but is largely based on the previous Zowe (1985). See also the somewhat similar review of Lemaréchal (1980).

1. Introduction

We will consider as a prototype problem the unconstrained minimization of a real function f:

$$\text{minimize } f(x) \text{ on } \mathbb{R}^n \tag{1.1}$$

where, in contrast to the standard situation, we do not require f to have continuous derivatives. More precisely: we are content if the gradient of f exists almost everywhere and if, at every point x where the gradient is not defined, at least the *directional derivative*

$$f'(x; d) := \lim_{t \downarrow 0} \frac{1}{t} [f(x + td) - f(x)] \tag{1.2}$$

exists in every direction d.

To simplify the presentation, we restrict most of our development to the case when f is a *convex function* from R^n to R (it is known that (1.2) then automatically holds). It is in this framework that things are easiest to explain. However, the theory can be extended to more general f with only technical changes.

Typically, the function f in (1.1) will be "piecewise-C^1", i.e. R^n will be composed of regions inside which the gradient ∇f exists and is continuous, and at the boundary of which ∇f jumps (although f itself is continuous). Consider, e.g. the function in one dimension:

$$f(x) := \begin{cases} -x & \text{for } x < 0 \,, \\ x^2 & \text{for } x \geqslant 0 \,. \end{cases} \tag{1.3}$$

Then $\nabla f(x) = -1$ for negative x and $\nabla f(x) = 2x$ for positive x. At $x = 0$ the gradient is not defined but, obviously, the two limits $\lim_{x \uparrow 0} \nabla f(x) = -1$ and $\lim_{x \downarrow 0} \nabla f(x) = 0$ taken together characterize the (first order) behaviour of f close to the kink $x = 0$. This leads us to the following substitution of the gradient: the *subdifferential* of f at x is (conv denotes the closed convex hull)

$$\begin{aligned} \partial f(x) := \operatorname{conv}\{ g \in \mathbb{R}^n \mid g = \lim \nabla f(x_i),\ x_i \to x, \\ \nabla f(x_i) \text{ exists, } \nabla f(x_i) \text{ converges} \} \,. \end{aligned} \tag{1.4}$$

This definition makes sense since, for convex f, the gradient exists almost everywhere. The subdifferential is a non-empty convex compact set which reduces to the gradient in case f is differentiable at x; the elements of $\partial f(x)$ are called *subgradients*. For the above f of (1.3) one gets: for $x \neq 0$, $\partial f(x)$ is the singleton $\nabla f(x)$, namely

$$\partial f(x) = \{\nabla f(x)\} = \begin{cases} -1 & \text{for } x < 0 \,, \\ 2x & \text{for } x > 0 \,, \end{cases} \tag{1.5}$$

while

$$\partial f(0) = [-1, 0] \,. \tag{1.6}$$

As expected, there is a close relation between the subdifferential and the directional derivative. Actually the directional derivative $f'(x;\ \cdot\ )$ is the *support function* of $\partial f(x)$, i.e.

$$f'(x; d) = \max_{g \in \partial f(x)} g^{\mathrm{T}} d \,. \tag{1.7}$$

It is easily checked, for example, that (1.7) together with (1.5) gives back (1.6).

The definition (1.4) is not the most classical one, but it lends itself to generalizations for nonconvex f. On the other hand, if f is convex, the differential quotient in (1.2) is monotonic in t; $f(x+d) \geqslant f(x) + f'(x, d)$ and this, together with (1.7), gives another equivalent characterization:

$$\partial f(x) = \{ g \in \mathbb{R}^n \mid g^{\mathrm{T}}(z-x) \leqslant f(z) - f(x) \text{ for all } z \in \mathbb{R}^n \} . \tag{1.8}$$

Finally, for fixed x, $f'(x, d)$ is convex in d. As such, it has a subdifferential at $d = 0$, which is precisely $\partial f(x)$.

It is important to understand what (1.8) says, in addition to (1.7): the latter is local and, loosely speaking, means

$$f(x + td) \geqslant f(x) + tg^{\mathrm{T}}d + \mathrm{o}(t) \quad \forall g \in \partial f(x)$$

while (1.8) is global and says that, for all $t \geqslant 0$, $\mathrm{o}(t)$ in the above estimate is nonnegative.

Properties (1.7) and (1.8) immediately give the *necessary and sufficient optimality condition* for the convex problem (1.1):

$$\begin{aligned} &x^* \text{ is optimal for (1.1) (i.e. } f(x^*) \leqslant f(x) \text{ for all } x) \\ &\quad \Leftrightarrow 0 \in \partial f(x^*) . \end{aligned} \tag{1.9}$$

Hence the set X^* of optimal points for (1.1) is characterized by

$$X^* = \{ x^* \in \mathbb{R}^n \mid 0 \in \partial f(x^*) \} .$$

To exclude pathological situations we will often assume that X^* is nonempty and bounded.

For the following, we will make the general assumption:

$$\text{At every } x \text{, we know } f(x) \text{ and one (arbitrary) } g \in \partial f(x) . \tag{1.10}$$

This assumption is actually fairly natural and a subgradient can usually be computed using only standard differential calculus (see Section 2 and more precisely Proposition 2.1). Loosely speaking: even if $\nabla f(x)$ does not exist, it does exist at some x_i "infinitely close" to x (see (1.4)) and this $g_i = \nabla f(x_i)$ is "infinitely close" to some $g \in \partial f(x)$. In practice, there will be a black box (a computer subprogram, sometimes called an oracle) which, given $x \in R^n$, answers $f = f(x)$ and $g \in \partial f(x)$. The situation will thus be similar to that in ordinary, smooth optimization, except that g will not vary continuously with x. For example, with f of (1.3), the black box could be

$$g := \begin{cases} -1 & \text{for } x < 0 , \\ 2x & \text{for } x \geqslant 0 . \end{cases}$$

By contrast, Section 4 will be devoted to problems for which the black box is more informative, namely

$$\text{At every } x, \text{ we know } f(x) \text{ and the full set } \partial f(x)\,. \tag{1.11}$$

To finish this section, we mention the lines along which the present theory is extended to the non-convex case: the basic tool is still definition (1.4) which, as already mentioned, makes sense when ∇f exists on a dense set. This is the case when f is *locally Lipschitzian* and, as such, has a gradient almost everywhere. Then $\partial f(x)$ is still a convex compact set, its support function still defines, through (1.7), a certain generalization of the directional derivative. In many applications the locally Lipschitzian f will be the result of some inner maximization, say $f(x) = \max_y h(x, y)$; in this special situation, the directional derivative (1.2) does exist and is given in terms of the subdifferential (1.4) by formula (1.7) (see Proposition 2.1); however, the global property (1.8) does not hold, then.

2. Examples of nonsmooth problems

Functions with discontinuous derivatives are frequent in operations research. Sometimes they arise already when modelling the problem itself, sometimes they are introduced artificially during the solution procedure. The latter appears as soon as one is concerned with any kind of *decomposition.*

2.1. Inherent nondifferentiability

Let x, a nonnegative variable, represent an income. In many economic systems, it induces a *tax* $T(x)$, which has discontinuous derivatives; a set of thresholds is given:

$$0 = a_0 < a_1 < \cdots < a_m = +\infty$$

together with rates $r_0, r_1, \ldots, r_{m-1}$, and T is given as follows:

$$\text{for } a_i \leqslant x < a_{i+1}, \quad T(x) := T_i + r_i x$$

with $T_0 := 0$, $T_i := T_{i-1} + a_i(r_{i-1} - r_i)$ (so T is continuous! it must even be a contraction, $|r_i| < 1$! it should also be increasing, $r_i \geqslant 0$! and even convex, $r_{i+1} > r_i$!). See Figure 2.1.

Remark 2.1. Convexity of T corresponds to the explicit expression

$$T(x) = \max\{T_i + r_i x \mid i = 0, \ldots, m-1\}\,, \tag{2.1}$$

which is easily checked. With relation to (1.1), convexity is a nice property in the present context since one usually wishes to minimize taxes; however, it is not the general rule in operations research: on the contrary, one often has to minimize concave functions (economies of scale).

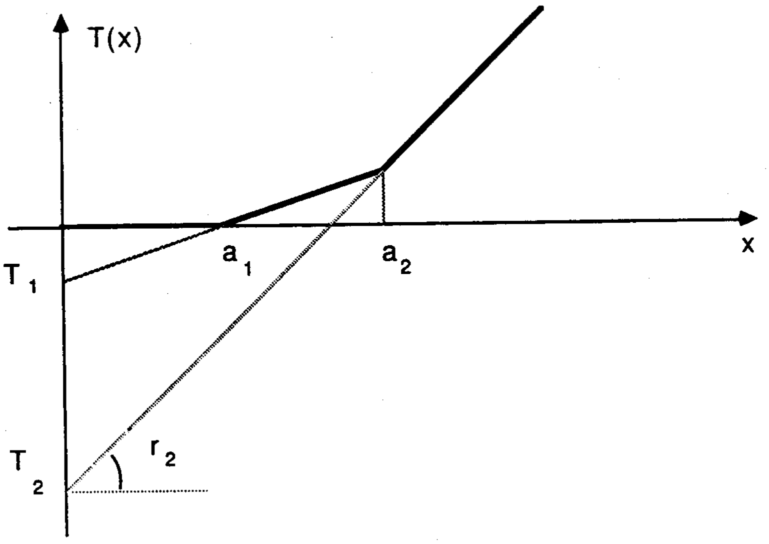

Fig. 2.1.

Expression (2.1) defines a class of functions which are frequently encountered in NSO, namely piecewise linear functions. More generally one can have functions like

$$f(x) = \max\{f_i(x) \mid i = 1, \ldots, m\} \tag{2.2}$$

where each f_i is smooth. Even more generally, one can have

$$f(x) = \max\{h(x, y) \mid y \in Y\} \tag{2.3}$$

where h is smooth with respect to x.

Minimizing a function given by (2.1), (2.2) or (2.3) is called the *minimax problem*. Note that computing f in (2.3) may be a time consuming task. These types of problems, however, are fully in the framework of assumption (1.10) thanks to the following result which states that g is available "for free" once f has been computed:

Proposition 2.1. *Let Y be a compact set, let $h(x, y)$ be a continuous function such that $\nabla_x h(x, y)$ is also continuous (jointly in x and y); then consider f defined by (2.3) and call*

$$M(x) := \{\nabla_x h(x, y) \mid y \text{ optimal at } x, \text{ i.e. } h(x, y) = f(x)\}$$

the set of "optimal gradients" at x. Then

(i) conv $M(x) = \partial f(x)$ *of* (1.4)

(ii) $f'(x, d) = \max\{g^T d \mid g \in M(x)\}$, *i.e.* (1.7) *holds.* □

Note, as a corollary, that f has a gradient whenever h is maximized at a single y, and differentiability usually fails when this y is no longer unique.

Functions of type (2.2) can be encountered in Tchebychef approximation, or also when the constraints of the optimization problem are treated through exact penalties. Note also, finally, that (2.3) is quasi equivalent to the so-called *semi-infinite programming* problem (T is an infinite set)

$$\min\ f(x) \quad \text{s.t.}\ c(x,t) \leqslant 0\ \forall t \in T$$

and both problems call for very similar methods.

2.2. Parametric decomposition

Suppose a given optimization problem has two (groups of) variables, x and y say, with the property that, for fixed x, minimizing with respect to y is very easy. An elementary instance of this situation is

$$\min c(x)^{\mathrm{T}} y\,, \tag{2.4a}$$

$$Ay = b\,, \tag{2.4b}$$

$$y \geqslant 0\,. \tag{2.4c}$$

Of course, it is then attractive to minimize with respect to y first, and then to minimize with respect to x the resulting function

$$f(x) := \min_y \{c(x)^{\mathrm{T}} y \mid Ay = b,\ y \geqslant 0\}\,. \tag{2.5}$$

Apply Proposition 2.1: at a given $\bar{x}$, call $y(\bar{x})$ the optimal polyhedron in (2.5). Provided $Y(\cdot)$ remains bounded in the neighborhood of $\bar{x}$, we have

$$\partial f(\bar{x}) = \{C'(\bar{x})^{\mathrm{T}} y \mid y \in Y(\bar{x})\}\,,$$

where C' is the Jacobian matrix of $c(\cdot)$ (observe that $C'^{\mathrm{T}} Y(\bar{x})$ is its own closed convex hull). Thus, computing the full ∂f implies the knowledge of *all solutions* of the linear program, while (1.10) amounts to finding one (arbitrary) solution.

Let us illustrate this point. Suppose (2.5) has a unique solution $\bar{y}$ at $\bar{x}$. For x close enough to $\bar{x}$, $\bar{y}$ is still the unique solution (see Figure 2.2), so $f(x) = c(x) \cdot \bar{y}$ and the variation of c alone gives the variation of f:

$$\nabla f(\bar{x}) = C_x'(\bar{x})^{\mathrm{T}} \cdot \bar{y}\,.$$

On the other hand, suppose there are several optimal y's in (2.5), i.e. $\bar{x}$ is a *kink*, where $\bar{y}$ jumps between optimal extreme points, say $\bar{y}$ and $\bar{y}_1$ on Figure 2.2; then ∇f jumps between $C'^{\mathrm{T}}\bar{y}$ and $C'^{\mathrm{T}}\bar{y}_1$, each of these two vectors being a valid gradient in some region of the x-space. All this is just what Proposition 2.1 says.

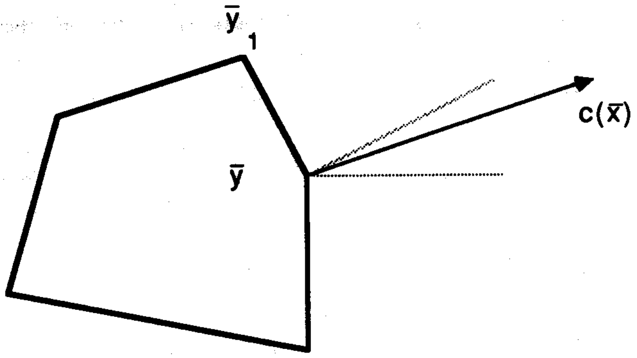

Fig. 2.2.

2.3. Decomposition by prices

Common optimization problems, in operations research, have the separable form

$$\min \sum_{i=1}^{n} f_i(x_i), \tag{2.6a}$$
$$\sum_{i=1}^{n} a_i(x_i) = 0 \tag{2.6b}$$

where each $a_i(x) \in R^m$, say. Forming the Lagrangian function (with $u \in R^m$)

$$L(x, u) := \sum [f_i(x_i) + u^{\mathrm{T}} a_i(x_i)] \tag{2.7}$$

and the dual function

$$q(u) := \min_x L(x, u) = \sum_i \min_{x_i} [f_i(x_i) + u^{\mathrm{T}} a_i(x_i)]$$

(obtainable via a minimization simpler than (2.6) because it is decomposed), the dual problem

$$\max \; q(u)$$

is sometimes useful for the solution of (2.6). Again, it is a problem of the form (1.1) and again, from Proposition 2.1, after computing $q(u)$ to obtain a primal variable $\bar{x}$, say, the constraint value $\Sigma\, a_i(\bar{x}_i)$ gives either $\nabla q(u)$ (if $\bar{x}$ is the unique minimizer of $L(u, \cdot)$) or a subgradient in $\partial q(u)$ (in fact, q is concave, so (1.8) holds for $-q$).

Remark 2.2. These properties are totally independent of any smoothness in (2.6). An extreme case is integer programming problems, where x in (2.6) is further constrained by

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, n .$$

Forming the Lagrangian function (2.7), we now define the dual function as

$$q(u) := \min L(x, u), \quad x \in \{0, 1\}^n,$$

which is trivial to compute, *no matter how complicated the functions f and a are* (it suffices to store the values $f_i(0)$, $a_i(0)$, $f_i(1)$, $a_i(1)$). Yet, the results of this section remain valid and q is as simple as possible, namely piecewise linear as in (2.1) (with $m = 2^n$, however). □

Iterating over u to solve (2.6) is sometimes called "Lagrangian relaxation" and we see that it fully belongs to the field of nondifferentiable optimization.

2.4. Decomposition by quotas (or allocation)

Another decomposition scheme can be used for (2.6). Take n vectors $y_i \in R^m$ and consider, for $i = 1, \ldots, n$,

$$v_i(y_i) := \min_z \{f_i(z) \mid a_i(z) = y_i\}. \tag{2.8}$$

Clearly, solving (2.6) just amounts to solving

$$\min \sum v_i(y_i),$$
$$\sum y_i = 0$$

which is essentially an unconstrained problem, but again non-smooth; differential properties of v_i are quite a technical subject; let us just mention that the subdifferential of v_i is given by Lagrange multipliers in (2.8), whenever this makes sense.

2.5. Stiff problems

An intuitive way to think of our kind of non-C^1 functions is to interpret them as C^2 functions with, at some points, very large eigenvalues in the Hessian matrix (then the gradient varies very rapidly).

Thus, smooth but badly conditioned problems can also be viewed as candidates for nonsmooth optimization. Although the theory of Section 1 does not apply, this view is still fruitful, in that it may help make classical methods more robust against large condition numbers. Section 6 below shows that *bundle methods*, precisely, consist in extending the theory of Section 1, so as to make it meaningful even for smooth f.

Remark 2.3. A conclusion of this Section 2 is that nonsmoothness is not reflected by any practical difficulty in computing gradients. All the examples above show that a nonsmooth function may be difficult to compute but, once this is done, a subgradient is readily obtained by applying usual differential calculus (Proposition 2.1). In this respect, and in view of (1.10) and Proposition 2.1, nonsmooth optimization is neither more nor less complicated than classical, smooth, optimization. □

3. Failure of smooth methods

3.1. Failure of convergence

In a classical smooth method one replaces f at x by a *linear or a quadratic model*

$$\nabla f(x)^{\mathrm{T}} d \quad [\simeq f(x+d)-f(x)]\,, \tag{3.1}$$

$$\nabla f(x)^{\mathrm{T}} d + \tfrac{1}{2} d^{\mathrm{T}} \nabla^2 f(x) d \quad [\simeq f(x+d)-f(x)]\,, \tag{3.2}$$

and one minimizes these models. Minimization of (3.1) on the unit ball gives the steepest descent and minimization of (3.2) gives Newton's method; compare formula (5.1) and (5.2), respectively. Obviously the above models are no longer defined at a kink and, close to a kink, they no longer provide an efficient approximation of f. If the minimal x^* is a kink (and this is almost the rule for non-smooth f), the search directions coming from either (3.1) or (3.2) become of little use when x approaches x^*.

As a result, the inappropriate smooth model (3.1) or (3.2) may cause convergence to a nonoptimal kink. Constructing examples to illustrate this statement is quite instructive. In R^2, consider the simple function

$$f_1(\xi, \eta) := 3(\xi^2 + 2\eta^2)$$

and apply the steepest descent algorithm, starting from $x_0 := (2, 1)$ (see the chapter on unconstrained optimization in this volume). A bit of calculation shows that $x_1 = (2, -1)/3$ and $x_2 = x_0/9$. Thus, the sequence $\{x_k\}$ generated by the steepest algorithm alternates between the two half-lines

$$H_{\pm} := \{(\xi, \eta) \mid 2\eta = \pm\, \xi\}$$

as shown on Figure 3.1, and tends to the optimal $x^* = (0, 0)$, as expected.

Now consider $f_2 := f_1^{1/2}$. Because its gradient is proportional to that of f_1, the same steepest descent algorithm generates the same $\{x_k\}$, which converges to the same $x^* = 0$, again optimal. Observe the new fact, however, that the corresponding gradient sequence $\{\nabla f_2(x_k)\}$ no longer converges to 0 but oscillates between the two fixed values $2^{1/2}(1, \pm 1)$.

Finally, construct a domain D containing the whole sequence $\{x_k\}$, for example

$$D := \{(\xi, \eta) \mid 0 \leq |\eta| \leq 2\xi\}$$

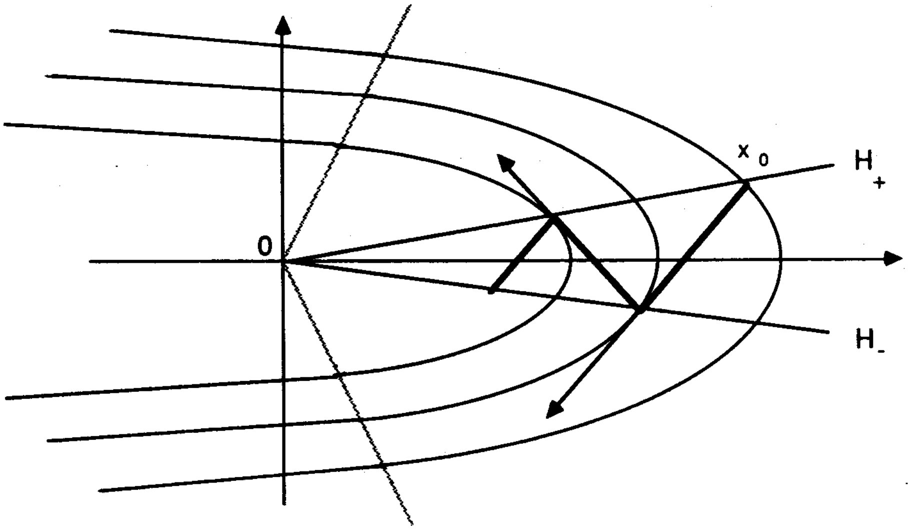

Fig. 3.1.

and modify f_2 out of D so that 0 is no longer optimal. For example extend the contours of f_2 by half-lines outside D (see Figure 3.1). The following function

$$f(\xi, h) := \begin{cases} [3(\xi^2 + 2\eta^2)]^{1/2} & \text{if } 0 \leq |\eta| \leq 2\xi\,, \\ (\xi + 4|\eta|)3^{-1/2} & \text{otherwise}\,, \end{cases} \tag{3.3}$$

does the job; it is convex, differentiable except on the left semi-axis, and "minimal" at $\xi = -\infty$. Notwithstanding, $\{x_k\}$ still converges to 0, totally ignoring the real behaviour of f.

Note that it is not difficult to perturb f_1 with a small non-quadratic term so that the same disaster happens (faster) with Newton's method. We can also make 0 the only kink in the space, or place the minimum of f where we want, etc. . . At any rate, the lesson of this example is that it is unwise to use classical Taylor models in nonsmooth optimization.

3.2. *Failure of optimality test*

Even without the pitfall of the above example, there remains a crucial handicap for a smooth method in a nonsmooth context. This is the *lack* of an implementable *stopping rule*. For a C^1-function the gradient will become small in norm, say

$$|\nabla f(x_k)| \leq \varepsilon \quad (\varepsilon > 0 \text{ small})\,, \tag{3.4}$$

when x_k approaches some optimal x^*; this can be used to stop the iterations automatically. However, for nonsmooth f a criterion like (3.4) does not make sense, even if the gradient existed at all iterates. E.g., for the function $f(x) := |x|$ we have $|\nabla f(x_k)| = 1$ in (3.4) at each $x_k \neq 0$, no matter how close x_k is to the optimal kink $x^* = 0$.

Considering definition (1.4), a stopping rule based on the optimality condition (which is now (1.9)) implies an exploration of the whole space neighboring the limit x^* we are interested in. Section 6.2 will exploit this fact.

3.3. Failure of gradient approximations

Another nasty property of nonsmooth functions is that computing subgradients is compulsory and finite differences are dangerous. For $x = (\xi, \eta, \phi) \in R^3$, consider the example $f(x) := \max\{\xi, \eta, \phi\}$. Its subdifferential at the origin is the unit simplex (apply Proposition 2.1):

$$\partial f(0) = \operatorname{conv}\{(1,0,0),\ (0,1,0),\ (0,0,1)\}\ .$$

On the other hand using forward, backward and central differences, we obtain respectively the points $(1,1,1)$, $(0,0,0)$ and $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ as candidates for an "approximate gradient"; none of them is close to $\partial f(0)$, i.e. none of them really represents the behaviour of f near x.

Actually, a mere finite differencing in nonsmooth optimization is a sin against mathematics. It is only because the differential $\delta f = \nabla f(x)^T \delta x$ varies linearly with the differential δx, in the smooth case, that it can be estimated by finite differences: a linear function is *uniquely determined* by its (finite set of) values on the canonical basis. No argument of that sort can be used in nonsmooth optimization, just because δf is *no longer linear* (see (1.7)): the differential δf must be estimated by $t^{-1}[f(x + td) - f(x)] = \delta f$ when $\delta x = td$ for all direction $d \in R^n$, and not only for a set of n directions.

Luckily, Remark 2.3 tells us that computing subgradients is usually not that difficult.

4. Special methods for special problems

In some problems, the nondifferentiability has a structure which makes them amenable to a classical, smooth, nonlinear programming problem. Basically, these problems are those where one *can construct* the full subdifferential $\partial f(x)$, and one *wishes to use* it numerically. Most of these problems have the form (2.2) (with m small). One can also encounter

$$f(x) = \sum_{i=1}^{m} |f_i(x)|\ . \tag{4.1}$$

A unified way to describe them all is to consider functions of the type

$$f(x) := h[C(x)] \tag{4.2}$$

where C maps R^n into R^m, each $C_i(x)$ is smooth while h maps R^m into R but is convex and nondifferentiable, like $|\cdot|_\infty$ or $|\cdot|_1$. For (4.2) to be amenable as above, one must assume that h itself is simple enough. The word "simple" roughly means: (4.2) would be easily solvable if C were linear. Of course, problems of this type are closer to nonlinear programming than to nonsmooth optimization; it is preferable to follow R. Fletcher in speaking of *composite problems*.

To solve problems of this class, one encounters first some *tricks*, generally aimed at smoothing f.

We give two examples:

For small $\varepsilon > 0$, one observes that

$$|f(x)| = (f^2(x))^{1/2} \simeq (f^2(x) + \varepsilon)^{1/2} \tag{4.3}$$

and this can help smoothing l_1-type functions. As for l_∞-type functions, one can observe that

$$\begin{aligned} \max\{f_1, \ldots, f_m(x)\} &= \max_u \left\{\sum u_i f_i(x) \,|\, u_i \geqslant 0, \sum u_i = 1\right\} \\ &\simeq \max_u \left\{\sum u_i f_i(x) - \tfrac{1}{2}\varepsilon \sum u_i^2 \,|\, u_i \geqslant 0, \sum u_i = 1\right\} =: f_\varepsilon(x) \,. \end{aligned} \tag{4.4}$$

Computing the maximal u in (4.4) amounts to projecting the vector $\varepsilon^{-1}[f_1(x), \ldots, f_m(x)] \in R^m$ onto the unit-simplex. This maximal u is unique, call it $u_\varepsilon(x)$, so the function $f_\varepsilon(x)$ is differentiable, with gradient given by Proposition 2.1:

$$\nabla f_\varepsilon(x) = \sum u_{\varepsilon i}(x) \nabla f_i(x) \,.$$

Of course the resulting approximating function (4.3) or (4.4), although smooth, will suffer from the usual bad conditioning of penalty-type functions.

The real operation underlying these smoothing techniques is a transformation of the nonsmooth problem into an equivalent smooth problem in which constraints usually appear. Let us illustrate this point on the finite minimax problem. Here, we will use the material of the chapter on nonlinear optimization in this volume. We start with the remark that

$$\min\ f(x) \quad \text{where } f(x) := \max\{f_1(x), \ldots, f_m(x)\} \tag{4.5}$$

is equivalent to the problem with $n + 1$ variables and m constraints

$$\min\ v \tag{4.6a}$$

$$v \geqslant f_i(x), \quad i = 1, \ldots, m \,. \tag{4.6b}$$

Associated with (4.6) is the Lagrangian function

$$v + \sum u_i(f_i(x) - v) \,.$$

Among the optimality conditions, there appears immediately

$$u_i \geqslant 0\,, \qquad \sum u_i = 1\,; \tag{4.7}$$

therefore, it makes sense to consider the ad hoc Lagrangian function,

$$L(x, u) := \sum u_i f_i(x) \tag{4.8}$$

whose connection with (4.5) is clear (compare (4.4)).

Now, the usual first-order optimality conditions for (4.6) are as follows: if x^*, v^* solves (4.6), then clearly $v^* = f(x^*)$ and there exist u_i^* satisfying (4.7) together with

$$\sum u_i^* \nabla f_i(x^*) = 0\,, \tag{4.9a}$$

$$u_i^* = 0 \quad \text{if } v^* > f_i(x^*)\,. \tag{4.9b}$$

This is nothing but the optimality condition (1.9) written for the problem (4.5), and using Proposition 2.1 to characterize $\partial f(x^*)$: setting

$$I(x) := \{i \mid f_i(x) = f(x)\}\,, \tag{4.10}$$

we have for optimal x^*

$$\exists u_i^*,\ i \in I(x^*) \quad \text{s.t. } u_i^* \geqslant 0,\ \sum u_i^* = 1,\ \sum u_i^* \nabla f_i(x^*) = 0\,. \tag{4.11}$$

Remark 4.1. Note the following explanation of Proposition 2.1, in relation with (1.7): take x and d in R^n; we have

$$f_i(x + td) = f_i(x) + t\nabla f_i(x)^{\mathrm{T}} d + \mathrm{o}_i(t)\,;$$

for $t > 0$ small enough, those i's not in $I(x)$ do not matter when computing $f(x + td)$, i.e.

$$\begin{aligned} f(x + td) &= \max\{f_i(x) + t\nabla f_i(x)^{\mathrm{T}} d + \mathrm{o}_i(t) \mid i \in I(x)\} \\ &= f(x) + t \max\{\nabla f_i(x)^{\mathrm{T}} d \mid i \in I(x)\} + \mathrm{o}(t) \end{aligned} \tag{4.12}$$

and formula (1.7) readily follows via convexification of the set

$$\{\nabla f_i(x) \mid i \in I(x)\}\,.$$

Second-order optimality conditions can also be derived for (4.6). First of all, to say that the gradients of the active constraints of (4.6)

$$\{\{-1, \nabla f_i(x^*)\} \mid i \in I(x^*)\}$$

are linearly independent is to say that the gradients of the active f_i's in (4.5)

$$\{\nabla f_i(x^*) \mid i \in I(x^*)\}$$

are affinely independent, i.e. u^* in (4.11) is unique. Now a vector orthogonal to the gradients of the active constraints at $\{v^*, x^*\}$ is a $\{w, d\}$ such that

$$\nabla f_i(x^*)^{\mathrm{T}} d = w \quad \forall i \in I(x^*) .$$

This implies $w = f'(x, d)$ (all the $\nabla f_i^{\mathrm{T}} d$ have the same, hence maximal, value); furthermore $w = 0$ (otherwise (4.11) could not hold) so such a d is characterized by

$$\nabla f_i(x^*)^{\mathrm{T}} d = f'(x^*, d) \;\; (=0) \quad \forall i \in I(x^*) . \tag{4.13}$$

As a result, the second-order optimality condition is: if the $\nabla f_i(x^*)$, $i \in I(x^*)$, are affinely independent, then, for d satisfying (4.13),

$$d^{\mathrm{T}} \sum u_i^* \nabla^2 f_i(x^*) d \geqslant 0 . \tag{4.14}$$

Let us interpret this: for d satisfying (4.13) and z arbitrary in R^n,

$$\begin{aligned} &f(x + td + t^2 z) - f(x) \\ &\quad \simeq t^2 \max\{\tfrac{1}{2} d^{\mathrm{T}} \nabla^2 f_i(x^*) d + \nabla f_i(x^*)^{\mathrm{T}} z \mid i \in I(x^*)\} . \end{aligned}$$

Therefore we obtain, fixing d in (4.13),

$$\forall z \in R^n, \;\; \exists i \in I(x^*) \quad \text{s.t.} \;\; \tfrac{1}{2} \alpha_i(d) + \nabla f_i(x^*)^{\mathrm{T}} z \geqslant 0$$

(where we have set $\alpha_i(d) := d^{\mathrm{T}} \nabla^2 f_i(x^*) d$). By a theorem of the alternative, this implies that there exists $u_i^* = u_i^*(d)$ satisfying (4.11) and

$$0 \leqslant \sum u_i^*(d) \alpha_i(d) = d^{\mathrm{T}} \sum u_i^*(d) \nabla^2 f_i(x^*) d . \tag{4.15}$$

It is interesting to note that this second-order condition, together with the first-order condition (4.11), is valid without any qualification condition. Of course, if (4.11) has a unique solution u^*, then $u_i^*(d)$ do not depend on d in (4.15) and we get back (4.14).

Accordingly, for x close to x^*, it is a good idea to minimize the function

$$\tilde{f}(d) := f(x) + \max\{\nabla f_i(x)^{\mathrm{T}} d \mid i \in I(x)\} + \tfrac{1}{2} d^{\mathrm{T}} H d \tag{4.16}$$

(where H approximates the matrix in (4.14)) which can be viewed as a satisfactory approximation to $f(x + d)$: the d-space can be decomposed in two subspaces:

– one, tangent to the constraints, where (4.13) holds; there, f is smooth and $\tilde{f}$

agrees with the Lagrangian function (4.8) which, because of (4.11) and (4.14), must be minimal at x^*;

– its orthogonal complement, where the first-order approximation $\tilde{f}$ of f is good enough: no second-order term matters because the first-order term is already positive.

From this point, the way is open to algorithms specially tailored for solving (4.6), or more generally (4.2): linearize C and add to h a quadratic term coming from the second order Taylor development along the kinky surface of h (i.e. where the nonsmooth nature of h does not play a role). From example, in the spirit of sequential quadratic programing to solve (4.6), the best idea to solve (4.5) is to define the direction-finding problem (x, standing for the current iterate x_k, is fixed, d is the variable)

$$\min_d \max_i [f_i(x) + \nabla f_i(x)^{\mathrm{T}} d] + \tfrac{1}{2} d^{\mathrm{T}} H d \tag{4.17}$$

where H is some quasi-Newton update for the Hessian of the Lagrangian function (4.8) (considering that v is a linear variable in (4.6), there is no reason to introduce any curvature along the v-axis!).

Methods of this Section 4 do not treat the problem in its full generality (1.10) and they do not really differ from classical methods for smooth optimization. For example, they do not fit with Section 2.5. On the other hand, they are worth studying because they are open to generalizations; for example, it is fruitful to use (4.16) as a basis for constructing methods coping with (1.10), even if the explicit use of all the underlying f_i's must eventually be given up.

5. Subgradient methods

5.1. Rationale

Suppose for the moment that f is smooth, say C^1 or C^2, at the current iterate x_k. In a standard first order method one makes a positive step t_k along the negative gradient (compare 3.1):

$$x_{k+1} := x_k - t_k \nabla f(x_k)\,, \tag{5.1a}$$

$$t_k > 0 \quad \text{(line search)}\,. \tag{5.1b}$$

The direction $-\nabla f(x_k)$ is a direction of descent; hence a line search along $x_k - t\nabla f(x_k)$, $t \geq 0$, will provide some $t_k > 0$ such that $f(x_{k+1}) < f(x_k)$. If we want to do better toward second-order, we may add some conditioner and multiply $\nabla f(x_k)$ by a matrix H_k (which is ideally close to the inverse of the Hessian of f at x_k, compare (3.2)), to obtain the scheme

$$\text{compute } H_k\,, \tag{5.2a}$$

$$x_{k+1} := x_k - t_k H_k \nabla f(x_k)\,, \tag{5.2b}$$

$$t_k > 0 \quad \text{(line search)}\,. \tag{5.2c}$$

For nonsmooth f the gradient at x_k may not exist, but it is clear what to do. By assumption (1.10) we know at least one subgradient at x_k; hence we will replace the gradient in (5.1) and (5.2), respectively, by a subgradient g_k. Normalizing the search direction we obtain as generalization of (5.1)

$$x_{k+1} := x_k - t_k g_k/|g_k| \quad \text{where } g_k \in \partial f(x_k)\,, \tag{5.3a}$$

$$t_k > 0 \quad \text{(suitable)}\,. \tag{5.3b}$$

The corresponding extension of (5.2) will be the subject of Sections 5.2 and 5.3.

How can we choose t_k in (5.3)?

Consider once more the function (3.3) at a point of nondifferentiability, say $x_k := (0, 0)$. By definition (1.4) the vector $g_k := 2^{1/2}(1, 1)$ is a subgradient at x_k. An inspection of the level lines in figure 3.1 tells us that this special $-g_k$ is *not a direction of descent*. There is no $t_k > 0$ such that $f(x_{k+1}) < f(x_k)$ with this g_k in (5.3). In contrast to (5.1), the steplength t_k cannot be determined via a line search. And note: even when the negative subgradient in (5.3) is a direction of descent, it is not advisable to make a line search, which could generate the disastrous steepest descent path of Figure 3.1. Nonsmoothness requires new ideas for the stepsize.

The following simple but basic observation shows us what to do: let x^* be optimal; from (1.8), the angle between $-g_k$ and $x^* - x_k$ is acute; hence, for $t > 0$ small enough $x_k - tg_k/|g_k|$ is closer to x^* than x_k. More precisely:

Lemma 5.1. *Suppose x_k is not optimal and let x^* be any optimal point. Then*

$$|x_{k+1} - x^*| < |x_k - x^*| \tag{5.4}$$

whenever

$$0 < t_k < 2[f(x_k) - f(x^*)]/|g_k|\,. \tag{5.5}$$

Proof. By definition (5.3), we have

$$\begin{aligned}|x^* - x_{k+1}|^2 &= |x^* - x_k + t_k g_k/|g_k||^2 \\ &= |x^* - x_k|^2 + 2t_k(x^* - x_k)^{\mathrm{T}} g_k/|g_k| + t_k^2 g_k^{\mathrm{T}} g_k/|g_k|^2\,,\end{aligned}$$

which we write as

$$|x^* - x_{k+1}|^2 = |x^* - x_k|^2 - 2t_k b_k + t_k^2 \tag{5.6}$$

with $b_k := (x_k - x^*)^{\mathrm{T}} g_k/|g_k|$. Then, (5.4) holds for t_k between 0 and $2b_k$.

Using (1.8) with $x = x_k$, $g = g_k$ and $z = x^*$ gives

$$b_k \geqslant [f(x_k) - f(x^*)]/|g_k| > 0\,, \tag{5.7}$$

so (5.5) implies the required property. □

Now let us choose the stepsize so that

$$t_k \downarrow 0 \quad \text{when } k \to +\infty .$$

This yields an argument for convergence to a true optimum: indeed, if (5.4) does not hold, then Lemma 5.1 implies that (5.5) does not hold, $f(x_k)$ is close to $f(x^*)$ (if k is large) and we are done.

There is still another item which has to be considered for choosing the sequence $\{t_k\}$, however: let x_0 be the starting point for iteration (5.3) and put $A := \Sigma_{j=0}^{\infty} t_j$. Then for each iteration index k,

$$\begin{aligned} |x_0 - x_k| &\leqslant |x_0 - x_1| + |x_1 - x_2| + \cdots + |x_{k-1} - x_k| \\ &= t_0 + t_1 + \cdots + t_{k-1} \leqslant A . \end{aligned} \tag{5.8}$$

In other words: we stay all the time in a ball with radius A around the starting point x_0. To be on the safe side, we choose the small t_k such that, neverthelesss, the sum of the t_k is large, say $+\infty$. Then also some optimal x^*, even far away from the starting point x_0, will not be out of reach. In summary, we obtain the following specification for (5.3):

$$\begin{aligned} & x_{k+1} := x_k - t_k g_k / |g_k| \quad \text{with } g_k \in \partial f(x_k) , \\ & t_k \text{ such that } t_k \downarrow 0 \text{ and } \sum_{k=0}^{\infty} t_k = \infty . \end{aligned} \tag{5.9}$$

Then, everything is set for the following result.

Theorem 5.1. *Suppose the set X^* of optimal points is nonempty and bounded. Then, for arbitrary starting point x_0, the sequence x_k provided by* (5.9) *is bounded and all its limit points are in X^*.* □

We omit the proof, which is rather technical.

Iteration (5.9) is of utmost simplicity; in particular, no line search is needed, of course. Unfortunately, one can expect only a poor convergence speed. Let us suppose for a moment that the x_k, given by (5.9), would tend to x^* *with geometric convergence rate* (also called *R-linear convergence*), i.e., there exists $M > 0$ and $0 < q < 1$ such that

$$|x_k - x^*| \leqslant M q^k \quad \text{for all } k .$$

Then, for all k,

$$t_k = |x_{k+1} - x_k| \leqslant |x_{k+1} - x^*| + |x^* - x_k| \leqslant M(q+1)q^k .$$

Summing up over k we would obtain

$$\sum_{k=0}^{\infty} t_k \leqslant M(q+1) \sum_{k=0}^{\infty} q^k = M(q+1)/(1-q) .$$

We get a contradiction since by choice of the t_k the left-hand side is $+\infty$. Consequently we have the disappointing supplement to the convergence result:

The convergence of the process (5.9) *is less than geometric.*

Geometric convergence does require $\{t_k\}$ to be a geometric sequence. Thus, an alternative to (5.9) is the following rule for the stepsize:

$$\text{choose } t_0 > 0,\ q \in]0,1[\text{ and take } t_k := t_0 q^k . \tag{5.10}$$

Then x_k converges geometrically to some limit $\bar{x}$ which, in view of (5.8), may not be optimal. It can be shown, however, that $\bar{x}$ is optimal when t_0 and q are larger than some (not computable) thresholds, respectively.

Finally, suppose we are in the following special situation:

$$\text{the optimal value of } f, \text{ say } f^*, \text{ is known} . \tag{5.11}$$

Then, instead of picking the stepsize blindly according to some *qualitative* rule (5.9) or (5.10), we can use Lemma 5.1 to control t_k in a *quantitative* way, namely:

$$\text{choose } \lambda \in]0,2[\text{ and take } t_k := \lambda[f(x_k) - f^*]/|g_k| . \tag{5.12}$$

This computable steplength, which incorporates the knowledge we have of f, x_k and X^*, guarantees a monotonic decrease of the distance from x_k to X^* (for all k) and actually the whole sequence $\{x_k\}$ does converge to some $x^* \in X^*$. As for the rate of convergence, various theorems can be proved, for example,

Theorem 5.2. *Suppose f has a minimizer x^*, the optimal value $f^* = f(x^*)$ is known and for some $l > 0$*

$$f(x) - f(x^*) \geq l|x - x^*| \quad \text{for all } x . \tag{5.13}$$

Then iteration (5.3) *with stepsize rule* (5.12) *provides a sequence converging to x^* with geometric convergence rate.* □

Remark 5.1. The function

$$f(x) := \max\{f_1(x), \ldots, f_m(x), 0\} ,$$

whose minimization is equivalent to the solution of the (assumed feasible) inequality system

$$\text{find } x \text{ such that } f_i(x) \leq 0 \quad (1 \leq i \leq m) ,$$

can serve as an example where one knows $f(x^*)$ (=0) without knowing x^*. For linear f_i's, iteration (5.3) with stepsize rule (5.12) is known as the relaxation method for solving the above inequality system.

5.2. Acceleration along the gradient

In case we neither know the optimal function value f^*, nor do we want to risk convergence to a non-optimal $\bar{x}$, then there remains only one way to accelerate the convergence of the subgradient method: we have to give up the Markov nature of iteration (5.3) by adding information obtained in previous steps. Essentially two approaches are known, both due to N.Z. Shor, which we study in Sections 5.2 and 5.3.

The first approach (dilation along the gradient) is based on the following idea: suppose, starting from x_0, we have found x_1 by a move along $-g_0$ and we have selected g_1 as the new subgradient at x_1. Then it makes sense to shorten that part of $-g_1$ which is collinear to the just used direction $-g_0$. Even more can be said: since the gradient direction is so notoriously bad – even in the smooth case – it is worthless to look for x^* (*i.e. to place the subsequent* x_k's) out of a region roughly orthogonal to g_0. Therefore let us dilate the space by a linear operator H_1 whose only difference from the unit matrix is to multiply g_0 by a coefficient smaller than 1: the dilation coefficient. If we cumulate the dilations of each iteration and write $\{H_k\}$ in a condensed form, then we obtain the following formulae:

$$d_k := -H_k g_k/(g_k^{\mathrm{T}} H_k g_k)^{1/2}, \qquad x_{k+1} := x_k + t_k d_k,$$
$$H_{k+1} := \alpha_k(H_k - \beta_k d_k d_k^{\mathrm{T}}) \tag{5.14a}$$

where α_k, β_k, t_k are suitable positive parameters and the initial matrix H_0 is positive definite, for example $H_0 = \alpha I$. For $\alpha_k \equiv 1$ and $\beta_k \equiv 0$ we get back (5.3). The choice $\beta_k = 1$ corresponds to a projection: g_k becomes a null-vector of H_{k+1}, and of all subsequent H_{k+i} as well.

One parameter choice proves to be especially interesting. Let n be the dimension of the space and choose α_k, β_k and t_k in (5.14a) as constants:

$$\alpha_k := n^2/(n^2 - 1), \quad \beta_k := 2/(n+1), \quad t_k := 1/(n+1). \tag{5.14b}$$

Then the following convergence result holds without further assumption on the convex f.

Theorem 5.3. *Suppose the starting point x_0 and the starting matrix $H_0 = \alpha I$ are such that $|x_0 - x^*|^2 \leq \alpha$ for some optimal x^*. Then there exists $M > 0$ and $q < 1$ such that for the sequence generated by* (5.14*a*,*b*):

$$\min_{0 \leq j \leq k} [f(x_j) - f(x^*)] \leq Mq^k \quad \textit{for all } k. \ \square \tag{5.15}$$

Thus, a subsequence of the function values $f(x_k)$ converges to $f(x^*)$ with R-linear convergence rate. A closer analysis shows that, for large n, $q \simeq 1 - 1/(2n^2)$.

Hence, unfortunately, q is close to 1, tending to 1 as n increases.

The method (5.14a,b) is nothing but the so-called ellipsoid method, recently popularized in a different context (see the Chapter on linear programming, in this volume).

5.3. *Dilation along the difference of gradients*

Despite their simplicity and their convergence properties, methods of Section 5.2 have a very disappointing numerical behaviour; one can say that the situation is about the same as in Section 5.1; compare the simplicity of iteration 5.9, and the generality of Theorem 5.1.

Another dilation is motivated by Figure 5.1. When f is really stiff, ie. when the angle between g_k and g_{k+1} is wide open (although x_k and x_{k+1} are close together) both directions $-g_k$ and $-g_{k+1}$ are bad and zig-zags appear; furthermore, the optimal x^* is likely to lie near the hyperplane

$$g_{k+1}^{\mathrm{T}}(x^* - x_{k+1}) = g_k^{\mathrm{T}}(x^* - x_{k+1}),$$

in which f is apparently kinky. Hence, it might be a good idea to dilate the space along the unwished direction $g_{k+1} - g_k$. The iteration formulae take on the form

$$d_k := H_k g_k, \tag{5.16a}$$

$$x_{k+1} := x_k + t_k d_k, \tag{5.16b}$$

$$H_{k+1} := H_k - \beta_k p_k p_k^{\mathrm{T}} / p_k^{\mathrm{T}}(g_{k+1} - g_k)$$
$$\text{where } p_k := H_k(g_{k+1} - g_k), \tag{5.16c}$$

compare (5.14a); we have absorbed the parameter α_k in the stepsize t_k. Here again, $\beta_k \equiv 0$ gives back the original formula (5.3).

Observe that the choice $\beta_k \equiv 1$ corresponds again to a projection along $g_{k+1} - g_k$, and then $d_{k+1}^{\mathrm{T}}(g_{k+1} - g_k) = 0$.

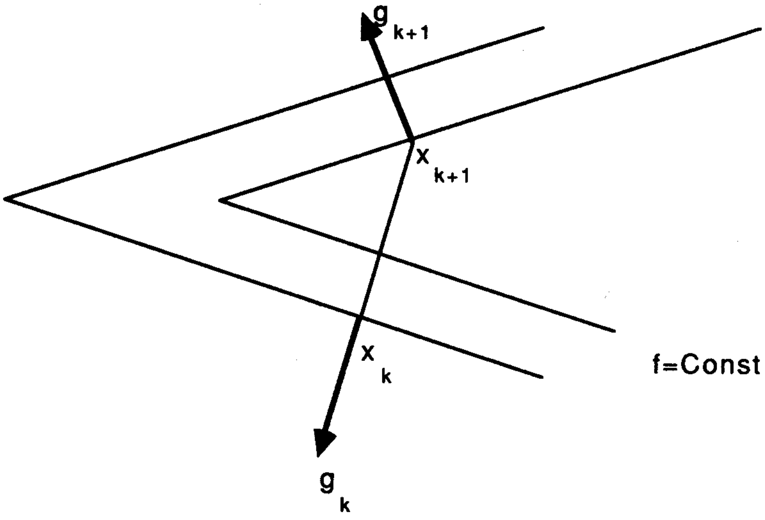

Fig. 5.1.

Here, however, no rule other than pure heuristics is known for controlling the dilatation parameter β_k and the stepsize t_k. It can be observed that, if f is quadratic (hence smooth) with minimum x^* and if t_k is optimal,

$$f(x_k + t_k d_k) \leqslant f(x_k + td_k) \quad \forall t > 0$$

then $(x^* - x_{k+1})^{\mathrm{T}}(g_{k+1} - g_k) = 0$, i.e. $\beta_k = 1$ is the best choice. Of course, this is of little use in the general case: whenever $\beta_k = 1$, $g_{k+1} - g_k$ becomes a null-vector of H_{k+i}, $i \geqslant 1$. Note also the similarity with quasi-Newton methods; for example, the Davidon–Fletcher–Powell update takes $\beta_k \equiv 1$ and adds the matrix

$$(x_{k+1} - x_k)(x_{k+1} - x_k)^{\mathrm{T}} / (x_{k+1} - x_k)^{\mathrm{T}}(g_{k+1} - g_k)$$

to H_{k+1} of (5.16) (see the chapter of unconstrained optimization, in this volume).

Despite very bad theoretical properties, method (5.16) has potentially very good numerical qualities.

5.4. Cutting planes

Suppose that, at iteration k, some set D_k is known to contain an optimal point x^*. After having computed x_{k+1} and $g_{k+1} \in \partial f(x_{k+1})$, we know from (1.8) that

$$x^* \in D_k \cap \{x \mid g_{k+1}^{\mathrm{T}}(x - x_{k+1}) \leqslant 0\} . \tag{5.17}$$

With this point of view in mind, it remains to place x_{k+1} (in D_k!).

The first idea has been to minimize the under-estimate

$$\hat{f}_k(x) := \max\{f(x_i) + g_i^{\mathrm{T}}(x - x_i) \mid i = 0, \dots, k\} \leqslant f(x) , \tag{5.18}$$

giving x_{k+1} as solution of the linear program

$$\min\ v , \tag{5.19a}$$

$$v \geqslant f(x_i) + g_i^{\mathrm{T}}(x - x_i) , \quad i = 0, \dots, k . \tag{5.19b}$$

This is the original cutting plane algorithm (note that – at least for small k – we may have to add a constraint $x \in D$ for some bounded D to guarantee a finite optimum in (5.19)).

Then one can easily prove:

Theorem 5.4. *Suppose there is some K such that* (5.19) *has a bounded solution at iteration K. Then* (5.19) *has an optimal solution for all $k \geqslant K$ as well, f has an optimal solution x^*, $\hat{f}_k(x_{k+1}) \uparrow f(x^*)$ and $f(x_k) \to f(x^*)$.* □

Because $\hat{f}_k$ is a piecewise linear approximation of f, it is reasonable to expect good convegence when f itself looks piecewise linear around x^*, i.e. when an assumption like (5.13) holds. This property, however, is incompatible with a smooth f, for which the optimality condition tells us that, at least for x close to x^*,

$$l|x-x^*|^2 \leq f(x) - f(x^*) \leq L|x-x^*|^2 \tag{5.20}$$

with $l \geq 0$ and L finite. For this case, the following holds.

Theorem 5.5. *Suppose f is C^2 and strongly convex, i.e.* (5.20) *holds with* $0 < l \leq L < +\infty$. *Then the cutting plane algorithm* (5.19) *converges geometrically with a ratio equivalent to* $1-(l/L)^n$ *when* $n \to \infty$. (*n is the dimension of the space*). □

Nonsmooth functions that are difficult to minimize rapidly are those for which the space neighbouring a given optimum x^* is divided in two regions, one where (5.13) holds, the other where it is (5.20). The following loose statement explains what we have in mind:

Claim 5.1.

- It is easy to minimize rapidly a function resembling $f(\xi, \eta) = \xi^2 + \eta^2$: take Newton's method.
- It is easy to minimize rapidly a function resembling $|\xi| + |\eta|$: use the algorithm given by (5.19).
- It is difficult to minimize rapidly a function resembling $\xi^2 + \eta^4$ (singular Hessian at the optimum).
- Exactly for the same reason, it is difficult to minimize rapidly a function resembling $|\xi| + \eta^2$ (general nonsmooth).

Considerations of this kind have given birth to a new school of optimization in which, instead of basing one's reasoning on local properties (like in Lemma 5.1., for example, or in all areas of smooth optimization) one uses global arguments (like in (5.17)). An optimization process is then considered as a *game* between the *algorithm* (which computes x_k) and the *black box* (1.10). For the algorithm, the game consists in obtaining good convergence properties against the worst possible black box. With this minimax point of view in mind, it is natural to place x_{k+1} with the sole help of "sure" information, like $x^* \in D_k$, ignoring any "dubious" information like (3.1), (3.2), or even $\hat{f}_k \simeq f$ in (5.18).

Then the apparently best idea is to define x_{k+1} as

$$x_{k+1} \textit{ is the center of gravity of } D_k\,.$$

The resulting algorithm, obtained by defining D_{k+1} from (5.17), is known as the method of *center of gravity*. With this method, the volume of D_k is divided at each iteration by a non-negligible factor, which allows the following result:

Theorem 5.6. *The method of centers of gravity converges linearly and its ratio is equivalent to* $1 - 1/(e-1)n$ *when* $n \to \infty$. □

Now a major result in this theory of "global algorithms" is as follows:

Theorem 5.7. *For any minimization method using only the information from the black box* (1.10), *there exists a convex function for which the method converges at best linearly with ratio* $q = \exp(-Y/n)$. □

In the above theorem, Y is a positive constant. Note that, when $n \to \infty$, the ratio q is thus equivalent to $1 - Y/n$.

As a result, if global algorithms are ranked according to the criterion "worst rate of convergence (independently of f) in the case of many variables", the conclusion is:

- the method of centers of gravity is qualitatively optimal, since its ratio behaves like $1 - 1/n$; unfortunately, one does not know an implementable way to compute the center of gravity of a polyhedron;
- the ellipsoid method (5.14a,b) is not bad, since its ratio behaves like $1 - 1/n^2$;
- the cutting plane method (5.19) is horrible: its ratio behaves like $1 - e^{-n}$, and depends on the function being minimized; furthermore, the complexity of an iteration k (the number of constraints in (5.19)) goes to infinity with k; for fairness, however, it must be said that the theoretical properties of this method have not been much studied; we add that the bundle methods of Section 6 are elaborated variants of (5.19).

Before closing this section, let us mention two more directions of present research, concerning the same theory.

First, observe that D_k is usually a polyhedron given by its faces:

$$D_k = \{x \mid g_i^{\mathrm{T}} x \le q_i,\ i = 0, \ldots, k\}. \tag{5.21}$$

For example, the optimal value of (5.19) can be shown to satisfy $v_{k+1} \le f(x_k)$; therefore, if we set

$$q_i := f(x_k) - f(x_i) + g_i^{\mathrm{T}} x_i, \quad i = 0, \ldots, k, \tag{5.22}$$

we see that any optimal x_{k+1} lies in the D_k thus defined by (5.21), (5.22).

Then, why not try to define "centers" which are easier to compute than the center of gravity? One example is the maximizer over D_k of the function

$$F_k(x) := \prod_{i \le k} (q_i - g_i^{\mathrm{T}} x) \tag{5.23}$$

or equivalently of Log F_k, which is concave. Of course, one cannot maximize F_k in a finite amount of time, but one can take x_{k+1} as the result of a limited number of Newton's iterations.

A function like F_k of (5.23) is sometimes called a *F-distance*, which is 0 on the boundary of D_k, positive inside. Note also that the cutting plane iterate solving (5.19) can also be called a center, maximizing the function

$$\min_{i \leqslant k} (q_i - g_i^{\mathrm{T}} x)$$

instead of (5.23).

The second idea consists in defining the "safeguard-polyhedron" like D_k in the graph space R^{n+1} instead of R^n. Clearly enough, any minimum pair $(f(x^*), x^*)$ lies in (compare (5.19))

$$D_k' := \{(v, x) \mid v \geqslant f(x_i) + g_i^{\mathrm{T}}(x - x_i),\ i \leqslant k;\ v \leqslant f(x_k)\}\,.$$

The x-part of a center (v_{k+1}, x_{k+1}) of D_k' is likely to be a better approximation of x^* than a center of D_k. In this respect, cutting plane is a clumsy idea, which places (v_{k+1}, x_{k+1}) on a *vertex* of D_k' (the lowest one).

6. Bundle methods

The rationale for methods in this Section is to force the decrease $f(x_{k+1}) < f(x_k)$ by all means, in contrast to the methods of Section 5.

6.1. A conceptual first order ε-descent method

Suppose for the moment that we know the whole subdifferential at the current iterate x_k; in Section 6.2 we will drop this assumption. Then, instead of performing a step in the direction of a randomly chosen subgradient (as in subgradient method), let us select that g_k in $\partial f(x_k)$ providing the steepest descent. The directional derivative $f'(x_k; d)$ being a measure for the descent we can expect (in a first order sense and for small t) along $x_k + td$, $t > 0$, we are led to the direction finding subproblem

$$\min_{|d| \leqslant 1} f'(x_k; d)\,. \tag{6.1}$$

The additional constraint $|d| \leqslant 1$ in (6.1) becomes necessary, since $f'(x_k; \cdot)$ is positively homogeneous. In case f is smooth one has $f'(x_k; d) = f'(x_k)d$, i.e., (6.1) is just the minimization of the classical *first order model* (3.1). We use (1.7) to rewrite (6.1) in the form

$$\min_{|d| \leqslant 1} \max_{g \in \partial f(x_k)} g^{\mathrm{T}} d\,. \tag{6.2}$$

The unit ball and $\partial f(x_k)$ are nonempty convex compact sets. Hence, by a well-known Minimax Theorem, (6.2) is equivalent to

$$\max_{g \in \partial f(x_k)} \min_{|d| \leq 1} g^T d\,. \tag{6.3}$$

For given g the minimizing d of length 1 is $-g/|g|$. In summary, for solving (6.1), we have to study the minimum-norm problem (which is uniquely solvable since $\partial f(x_k)$ is a nonempty closed convex set)

$$\min_{g \in \partial f(x_k)} |g|\,. \tag{6.4}$$

If and only if g_k is the subgradient at x_k closest to the origin, then $d_k := -g_k/|g_k|$ minimizes $f'(x_k; \cdot)$ and one has $f'(x_k; d_k) = -|g_k|$. Because of (1.9), $g_k \neq 0$ for nonoptimal x_k and thus $f'(x_k; d_k) < 0$, i.e. (6.1) does provide a descent direction as long as there is one. We are led to the iteration scheme

$$d_k := -g_k/|g_k| \quad \text{with } |g_k| = \min_{g \in \partial f(x_k)} |g|\,, \tag{6.5a}$$

$$x_{k+1} := x_k + t_k d_k \quad \text{with } f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + td_k)\,. \tag{6.5b}$$

In case the gradient exists at x_k, the subdifferential is $\{\nabla f(x_k)\}$ and (6.5) reduces to the classical steepest descent method (with exact line search). Hence the example discussed in Section 3.1 warns us that for nonsmooth f the above iteration may collapse close to a kink. It is important to understand the reason for this failure and to develop a feeling of what could serve as safeguard in (6.5) against non-convergence in the nonsmooth situation. For this purpose consider once more the function (3.3) and the path shown in Figure 3.1. Going back to definition (1.4) we realize that the limits

$$2^{1/2}(1,1) = \lim \nabla f(x_{2k}) \quad \text{and} \quad 2^{1/2}(1,-1) = \lim \nabla f(x_{2k+1})$$

belong to the subdifferential of f at the origin. A more detailed (but straightforward) analysis shows that $\partial f(0)$ is the part of the ellipsis displayed in Figure 6.1:

$$\partial f(0) = \{(\xi, \eta) \,|\, 2\xi^2 + \eta^2 \leq 6;\ \xi \leq 3^{-1/2}\}\,.$$

Consequently iteration (6.5) would use at $x_k := 0$ the excellent direction $d_k = -(1,0)$.

Unfortunately, 0 is never reached: Figure 3.1 shows that we walk in shorter and shorter steps on a zigzagging path along the two directions

$$d' := d_{2k} = (-1,-1)2^{-1/2}\,, \tag{6.6a}$$

$$d'' := d_{2k+1} = (-1,1)2^{-1/2}\,. \tag{6.6b}$$

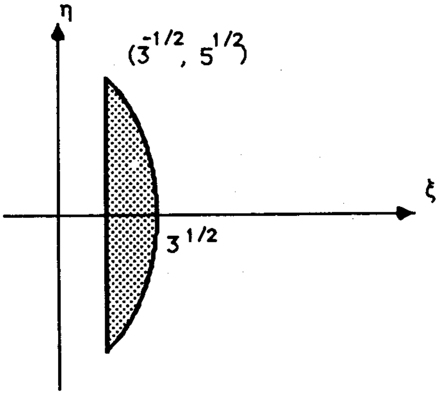

Fig. 6.1.

In a strictly local sense d' or d'' provides the steepest descent at x_k, whereas in a more global sense d' and d'' are less and less profitable. The more we approach 0, the poorer the (sub)gradient information becomes, due to the nondifferentiability at 0. This suggests enriching the information at x_k and replacing $\partial f(x_k)$ in (6.5) by $\bigcup_{y \in B} \partial f(y)$, where B is a suitable neighborhood of x_k. Then for x_k close to its limit 0, the enlarged set $\bigcup_{y \in B} \partial f(y)$ will contain $\partial f(0)$, i.e., the above ideal direction $(-1, 0)$ will be a candidate for the line search and we will escape. Luckily we have at hand an object in convex analysis which helps to realize what we have in mind. Fix some small $\varepsilon > 0$ and add this ε in the inequality defining the subdifferential (we attach ε as a subscript),

$$\partial_\varepsilon f(x) := \{g \in \mathbb{R}^n \mid g^{\mathrm{T}}(z - x) \leq f(z) - f(x) + \varepsilon \text{ for all } z \in \mathbb{R}^n\} . \tag{6.7}$$

This so-called *ε-subdifferential*, whose elements are called *ε-subgradients*, is again a nonempty convex compact set (same proof as for $\varepsilon = 0$), and the basic auxiliary result holds:

Lemma 6.1. *For every x and every $\varepsilon > 0$ there exists a neighborhood B of x such that*

$$\partial_\varepsilon f(x) \supset \bigcup_{y \in B} \partial f(y) .$$

Proof. Let x and $\varepsilon > 0$ be given. For $p \geq 0$, denote by B_p the ball of radius p around x. Let L be a Lipschitz constant of f on B_1; set

$$p := \min(1, \varepsilon/2L) , \quad B := B_p ,$$

let $y \in B$ and $g \in \partial f(y)$ be given. We prove $g \in \partial_\varepsilon f(x)$. Indeed, from the subgradient inequality (1.8):

$$g^{\mathrm{T}}(z - y) \leq f(z) - f(y) \quad \text{for all } z \in R^n ,$$

which we write

$$g^{\mathrm{T}}(z-x)+g^{\mathrm{T}}(x-y)\leq f(z)-f(x)+f(x)-f(y)\,;$$

this implies

$$g^{\mathrm{T}}(z-x)\leq f(z)-f(x)+|f(x)-f(y)|+|g|\cdot|x-y|\,.$$

Now obersve that $y\in B\subset B_1$. Therefore we obtain, using the Lipschitz condition

$$\begin{aligned}g^{\mathrm{T}}(z-x)\leq f(z)-f(x)+2L|x-y|&\leq f(z)-f(x)+2Lp\\&\leq f(z)-f(x)+\varepsilon\,.\end{aligned}$$

Thus, g satisfies the defining inequality (6.7) and the lemma is proved. □

Lemma 6.1 shows that the set $\partial_\varepsilon f(x_k)$ contains in a condensed form the subgradient information from a whole neighbourhood of x_k. As we will see, this will help to overcome the shortcoming caused by nonsmoothness.

To become more precise, fix some small $\varepsilon>0$ throughout the following and replace $\partial f(x)$ by $\partial_\varepsilon f(x)$. If we add the same ε in the definition of the directional derivative (for convex f the operation lim in (1.2) can be replaced by inf)

$$f'_\varepsilon(x;d):=\inf_{t>0}\frac{1}{t}\,[f(x+td)-f(x)+\varepsilon]\,, \tag{6.8}$$

then this *ε-directional derivative* again is the support function of the ε-subdifferential (same proof as for $\varepsilon=0$):

$$f'_\varepsilon(x;d)=\max_{g\in\partial_\varepsilon f(x)} g^{\mathrm{T}}d$$

(compare (1.7)). The arguments used at the beginning of this Section 6.1 show that also the ε-modified problems

$$\min_{|d|\leq 1} f'_\varepsilon(x_k;d) \tag{6.9}$$

and

$$\min_{g\in\partial_\varepsilon f(x_k)} |g| \tag{6.10}$$

correspond to each other. In words: if g_k solves (6.10), then $d_k:=-g_k/|g_k|$ minimizes $f'_\varepsilon(x_k;\cdot)$ and $f'_\varepsilon(x_k;d_k)=-|g_k|$. As long as $0\notin\partial_\varepsilon f(x_k)$ the minimal g_k in (6.10) is nonzero and thus $f'_\varepsilon(x_k;d_k)<0$. Definition (6.8) implies that a move along d_k guarantees a decrease of at least ε:

$$0 \notin \partial_\varepsilon f(x_k) \quad \Rightarrow \quad f(x_k + td_k) < f(x_k) - \varepsilon \text{ for suitable } t > 0 . \tag{6.11}$$

Now suppose $0 \in \partial_\varepsilon f(x_k)$. Put $g = 0$ in definition (6.7) to realize that x_k is already "almost" optimal; more precisely:

$$0 \in \partial_\varepsilon f(x_k) \quad \Rightarrow \quad f(x_k) \leq f(x) + \varepsilon \text{ for all } x . \tag{6.12}$$

Taken together, (6.11), (6.12) motivate the following ε-modification of iteration (6.5):

$$d_k := -g_k/|g_k| \quad \text{with } |g_k| = \min_{g \in \partial_\varepsilon f(x_k)} |g| , \tag{6.13a}$$

$$x_{k+1} := x_k + t_k d_k \quad \text{with } f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + td_k) . \tag{6.13b}$$

Obviously any pathological behaviour as in Figure 3.1 is now excluded. Indeed, if $\inf f(x) > -\infty$, then in finitely many steps one must reach some x_k such that $0 \in \partial_\varepsilon f(x_k)$, i.e., x_k is *ε-optimal*. Hence $0 \in \partial_\varepsilon f(x_k)$ serves as a stopping criterion. We summarize:

Theorem 6.1. *Let* $f^* := \inf f(x)$ *and let* $\{x_k\}$ *be the sequence generated by* (6.13) *for arbitrary starting point* x_0.

(a) *If* $f^* = -\infty$ *then* $\lim_{k \to \infty} f(x_k) = -\infty$.

(b) *If* $f^* > -\infty$ *then there exists k such that* $f(x_k) \leq f^* + \varepsilon$. □

Before we discuss an implementation of the above idea let us give another explanation for the change caused in the convergence behaviour by the transition from (6.5) to (6.13). The disaster in Figure 3.1 stems from the discontinuity of the subproblem in (6.5):

$$x \to d(x) := -g(x)/|g(x)| \quad \text{where } |g(x)| = \min_{g \in \partial f(x)} |g| .$$

Although the x_k converge to 0, the directions (compare with (6.6))

$$d(x_{2k}) = d' \quad \text{and} \quad d(x_{2k+1}) = d''$$

do not converge to $d(0) = (-1, 0)$. Using a result about the Lipschitz continuity of the point-to-set mapping $x \to \partial_\varepsilon f(x)$, one can easily verify that the slightly modified mapping in (6.13)

$$x \to d_\varepsilon(x) := -g_\varepsilon(x)/|g_\varepsilon(x)| \quad \text{where} \quad |g_\varepsilon(x)| = \min_{g \in \partial_\varepsilon f(x)} |g|$$

becomes continuous. Thus for x_k close to 0 the directions $d(x_k)$, used in (6.13), will be sufficiently close to the ideal direction $d(0) = (-1, 0)$ and we will make a move away from x_k, overtaking 0.

6.2. Implementation

Let us come back to the general situation (1.10) where at x_k we know but one subgradient g_k. Despite this minimal information, the bundle idea realizes (6.13) in a rather sophisticated way. The basic idea of a *bundle-type-algorithm* consists in replacing $\partial_\varepsilon f(x_k)$ by some inner approximating polytope P (*forming of the bundle*) and in solving (6.13) with $\partial_\varepsilon f(x_k)$ replaced by P. Provided P is a sufficiently good approximation, then we will find a direction along which a line search yields some x_{k+1} with a decrease of almost ε. In case P is a bad approximation (the line search will let us know this), then we stay at x_k and try to improve the approximating P by adding a further subgradient (so-called *nullstep*). These two items are the crucial ingredients of all bundle methods. We will try to describe in very short terms the two mentioned basic steps by avoiding all technical details (which, of course, are very important for an efficient implementation); in particular we will not specify what the above "almost ε" means.

Let $g_j \in \partial f(x_j)$, $j = k-1, k-2, \ldots$, be a collection of subgradients already computed. It is a trivial exercise to show that each g_j is a p_j-subgradient at x_k, where

$$p_j := f(x_k) - f(x_j) - g_j^{\mathrm{T}}(x_k - x_j) \tag{6.14}$$

(proceed as in the proof of Lemma 6.1). Then it is not difficult to choose P, for example

$$P := \left\{ \sum \lambda_j g_j \mid \lambda_j \geqslant 0,\ \sum \lambda_j = 1,\ \sum \lambda_j p_j \leqslant \varepsilon \right\} \tag{6.15}$$

which can be proved to be contained in $\partial_\varepsilon f(x_k)$. Note that $\varepsilon \geqslant 0$ is a free parameter that can be controlled, depending on one's optimism: a small ε hopefully gives a good approximation of $\partial_\varepsilon f(x_k)$ while a large ε corresponds to a large expected decrease in f.

Replacing $\partial_\varepsilon f(x_k)$ by P in (6.13), the determination of a search direction reduces to a quadratic programming problem:

$$d = -\mathrm{argmin}\{|g|^2 / g \in P\}\ . \tag{6.16}$$

In case P provides a good approximation of $\partial_\varepsilon f(x_k)$, then the resulting direction yields a decrease of almost ε and we are done. Now suppose that, on the contrary, P is a bad approximation. Then it may happen that the substitute problem (6.16) provides a direction d which is not even a descent direction; recall that not every negative subgradient guarantees a decrease. Then we compute a further subgradient g_+ at $x_k + td$ for some small $t > 0$. By Lemma 6.1, g_+ is an ε-subgradient at x_k. On the other hand, because $g_+ \in \partial f(x_k + td)$ and because d is bad, we have essentially

$$g_+^{\mathrm{T}}[x - (x + td)] \leqslant f(x) - f(x + td) \leqslant 0\ ,$$

i.e. $g_+^T d \geq 0$. As suggested by Figure 6.2, this implies that the polytope

$$P_+ := \operatorname{conv}(P \cup \{g_+\}) \subset \partial_\varepsilon f(x_k)$$

is a definitely better approximation than P. Hence, it remains to solve (6.16) with this new P_+ and to do the next line-search starting from the same x_k. This is a *null-step*.

The key property of this mechanism, namely that P_+ is definitely larger than P, has an analytical formulation:

Theorem 6.2. *If infinitely many null-steps are performed, then the solution of* (6.16) *tends to* 0. □

The proof of this statement, not particularly interesting, can be replaced by a look at Figure 6.2. As a result, when $|d|$ and ε are small, then x_k is approximately optimal (see (6.12): the 0-vector is almost in $\partial_\varepsilon f(x_k)$); to say that $|d|$ does not become small, on the other hand, means that, at some stage, the process terminates with a downhill d. This is what happens when 0 is far from $\partial_\varepsilon f(x_k)$.

6.3. *The bundle algorithms*

Exploiting all these ideas, one obtains the following schematic algorithm: start from $x_1 \in R^n$, $g_1 \in \partial f(x_1)$, set $p_1 = 0$. At each iteration k:

a. Compute the direction: $d_k = -\Sigma_{j=1}^k \lambda_j g_j$ where $\lambda \in R^k$ solves (compare (6.14), (6.15); ε is a control parameter)

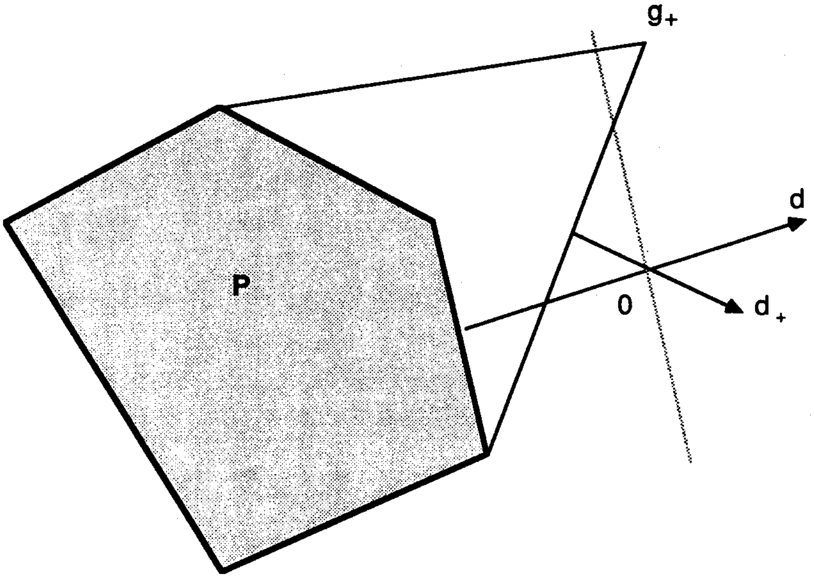

Fig. 6.2.

$$\min \frac{1}{2}\left|\sum \lambda_j g_j\right|^2 , \tag{6.17a}$$
$$\lambda_j \geq 0 , \tag{6.17b}$$
$$\sum \lambda_j = 1 , \tag{6.17c}$$
$$\sum \lambda_j p_j \leq \varepsilon . \tag{6.17d}$$

b. Check for stopping: if $|d_k|$ is small then quit or [reduce ε and go to a].

c. Do the line-search: find $t > 0$, $g_{k+1} \in \partial f(x_k + td_k)$ with $g_{k+1}^{\mathrm{T}} d_k$ large enough, yielding
either a descent step: $f(x_k + td_k)$ small enough; then $x_{k+1} = x_k + td_k$;
or a null step: $p_{k+1} := f(x_k) - f(x_k + td_k) + tg_{k+1}^{\mathrm{T}} d_k \leq \varepsilon$; then $x_{k+1} = x_k$.

d. Update the weights: in case of a descent step set $p_{k+1} = 0$ and replace each p_j by

$$p_j + f(x_{k+1}) - f(x_k) - tg_j^{\mathrm{T}} d_k \, ;$$

replace k by $k+1$ and loop to a.

Two details (among others) have been neglected in the above description. One is that, despite the appearances to the contrary, the size of (6.17) needs not grow indefinitely with k: after solving (6.17) one can set $g_0 = -d_k$ and $p_0 = \Sigma_{j=1}^k \lambda_j p_j$ because g_0 is a p_0-subgradient at x_k; then this p_0 is updated at step d and all convergence proofs remain valid if the $(k+1)$st problem (6.17) contains just the two elements (p_0, g_0) and (p_{k+1}, g_{k+1}).

The second detail concerns the line-search: what does it mean to require "$g_{k+1}^{\mathrm{T}} d_k$ large enough" and "$f(x_k + td_k)$ small enough". After solving (6.17), let $s \geq 0$ be the multiplier associated with the last constraint, i.e. (6.17) is equivalent to (compare Section 2.3)

$$\min \frac{1}{2}\left|\sum \lambda_j g_j\right|^2 + s \sum \lambda_j p_j , \tag{6.18a}$$
$$\lambda_j \geq 0 , \quad \sum \lambda_j = 1 . \tag{6.18b}$$

Examining the optimality conditions, one can see that

$$v_k := -|d_k|^2 - s\varepsilon \leq -|d_k|^2 < 0$$

can be considered as an estimate of $f'(x_k, d_k)$. Then, choosing $0 < m_1 < m_2 < 1$,

$g_{k+1}^{\mathrm{T}} d_k$ large enough means $g_{k+1}^{\mathrm{T}} d_k \geq m_2 v_k$,

$f(x_k + td_k)$ small enough means $f(x_k + td_k) \leq f(x_k) + m_1 t v_k$,

in the spirit of what is done in classical, smooth, optimization.

We mention that ε for step a. should not be thought of as a small parameter (which would give basically the steepest descent method, whose convergence is so bad, even in the smooth case; recall Section 2.5). Quite the contrary, the difficulty in this algorithm is that ε should be suitably chosen at each iteration, and this is crucial for fast convergence. Instead of choosing ε, one could choose s and solve (6.18); but, s is probably even more difficult to choose empirically than ε (at least, ε is homogeneous to f-values, but what about s?). However, (6.18) provides a link with Sections 4 and 5. We begin with the latter and consider again (5.18). To simplify notations, assume x_k is the best iterate: $f(x_k) \leqslant f(x_i)$, $i \leqslant k$. Minimizing $\hat{f}_k$ to find x_{k+1} makes sense only if $\hat{f}_k \simeq f$ *near* x_k (i.e. where the optimum is likely to lie); but there are good reasons to think that this is not the case and that $\hat{f}_k$ is far too optimistic (although $\hat{f}_k(x_i) = f(x_i)$, $i \leqslant k$); one of these reasons is the so slow convergence observed for the cutting plane algorithm. Therefore, it may appear as a sensible idea to minimize $\hat{f}_k(x) + \frac{1}{2}s|x - x_k|^2$ (for some chosen s) which is less optimistic than $\hat{f}_k$. Furthermore, use (6.14) as a notation to write $\hat{f}_k$ as

$$\hat{f}_k(x_k + d) = f(x_k) + \max\{-p_j + g_j^{\mathrm{T}}d \mid j \leqslant k\}\,. \tag{6.19}$$

In summary we obtain

$$\min\ v + \tfrac{1}{2}s|d|^2\,, \tag{6.20a}$$

$$v \geqslant -p_j + g_j^{\mathrm{T}}d\,, \tag{6.20b}$$

which is just (5.19) when $s = 0$.

Applying duality theory to this problem (see Section 2.3) we obtain now

$$x_{k+1} = x_k - s^{-1} \sum \lambda_j g_j \quad \text{with } \lambda \text{ solving (6.18)}\,.$$

Thus, a bundle method appears as a stabilization of the cutting plane algorithm, and also as a method of centers in the graph space (see the very end of Section 5.4). Note, however, that this does not help in choosing s in (6.18) or ε in (6.17).

The link with Section 4, now, will give an interpretation for the additional term $s|d|^2$. To make this link more suggestive, consider instead of P of (6.15), the coarser

$$P' := \operatorname{conv}\{g_j \mid p_j \leqslant \varepsilon\} \subset P\,.$$

Then the corresponding projection problem (6.17) or (6.18) becomes

$$\min \frac{1}{2}\left|\sum \lambda_j g_j\right|^2\,, \quad \lambda_j \geqslant 0,\ \sum \lambda_j = 1\,, \tag{6.21}$$

which can be interpreted as: to compute the direction, take only those g_j that are close to $\partial f(x_k)$, and pretend that they are in $\partial f(x_k)$ (actually, the first bundle methods, known as *conjugate subgradient methods*, were based on this idea).

Now apply again duality theory to (4.16): $\tilde{f}$ is minimized at $d = -H^{-1}[\Sigma\ \lambda_i \nabla f_i(x_k)]$ where $\lambda \in R^{I(x)}$ solves

$$\min\ \frac{1}{2}\left[\sum \lambda_i \nabla f_i(x)\right] H^{-1}\left[\sum \lambda_i \nabla f_i(x)\right], \quad \lambda_i \geq 0,\ \sum \lambda_i = 1\,,$$

whose analogy with (6.21) is clear.

Thus, we obtain the further interpretation of bundle methods: at iteration k, we are (temporarily) faced with the minimax function $\hat{f}_k$ of (6.19). Of this function, only those pieces

$$f(x_k) - p_j + g_j^{\mathrm{T}} d \quad \text{with } p_j \text{ small}$$

are reliable (see above). Neglecting the small p_j, we can consider that those pieces are linear approximations of some functions f_j, which in turn induce a curvature in some Lagrangian function. In bundle methods, this Hessian is taken as $H = sI$, for want of a better approximation. Finally, going from P' to P, i.e. from (6.21) to (6.18), is the same as going from (4.16) to (4.17). It amounts to using the pieces neglected in (6.21) as further safeguards to help computing d.

Our final comment about bundle methods is that they can be extended to more general f without conceptual difficulty. The key idea is to take Lemma 6.1 as a *definition* of $\partial_\varepsilon f(x)$ for nonconvex f:

$$\partial_\varepsilon f(x) = \operatorname{conv} \cup \{\partial f(y) \mid |y - x| \leq \varepsilon\}\,,$$

which amounts to changing p_j in (6.14) to $p_j = |x_k - x_j|$.

7. Directions for future developments

For smooth f and $\varepsilon \to 0$ iteration (6.13) reduces to the classical steepest descent method which is not more than linearly convergent. To obtain faster convergence one has to replace the first order model (3.1) behind the steepest descent idea by the second order model (3.2). Minimization of (3.2) leads to (Quasi-) Newton methods of type (5.2) which are at least superlinearly convergent and consequently behave much better in practice. There arises the natural and challenging question in NSO: Can we develop a "second order" model for nonsmooth f, based on the ε-modifications of standard definitions from convex analysis introduced in Section 6, i.e., a model which for smooth f and $\varepsilon \to 0$ reduces to (3.2)? The minimization of such a model should lead to much better convergence behaviour.

We will mention two possible approaches toward this aim.

7.1. Directional second order approximation of f

We start with the following proposal for a *second directional derivative* at x in direction d (provided the limit exists)

$$f''(x;d) := \lim_{t\downarrow 0} \frac{1}{t}\,[f'(x+td;d) - f'(x;d)]\,. \tag{7.1}$$

For $C^2 f$ the Mean-Value Theorem implies

$$\frac{1}{t}\,[f'(x+td;d) - f'(x;d)] = \frac{1}{t}\,[\nabla^2 f(x+\theta td)td]^{\mathrm{T}} d \quad \text{with } \theta \in (0,1)\,;$$

taking the limit we see that $f''(x;d)$ reproduces the Hessian:

$$f''(x;d) = d^{\mathrm{T}} \nabla^2 f(x) d\,. \tag{7.2}$$

We remark that, contrary to $f'(x;d)$, the above limit does not exist for every convex f; the function $f(x) := |x|^\alpha$ with $\alpha \in (1,2)$ and $x = 0$ may serve as a counter example.

Provided $f''(x;d)$ exists we can define the model

$$f'(x;d) + \tfrac{1}{2} f''(x;d)\,. \tag{7.3}$$

For C^2 f (7.3) is identical with (3.2) and thus minimization of (7.3) leads to Newton's method. This looks rather promising. However, we still feel uneasy. Firstly, in view of Figure 3.1, an ε is missing somewhere in (7.3); as we know from Section 6.1, such ε helps to avoid shortcomings due to nonsmoothness. Secondly, we do not dispose of a max-expression of type (1.7) for $f''(x;d)$. Such a relation, however, is important for an eventual implementation of (7.3); compare Section 6.2. The following idea proves to be helpful. Consider another limit (provided it exists)

$$c(x;d) := \lim_{\varepsilon\downarrow 0} \frac{f'_\varepsilon(x;d) - f'(x;d)}{\varepsilon^{1/2}}\,. \tag{7.4}$$

A straightforward geometric argument shows that the infimum $f'_\varepsilon(x;d)$ is the slope of the line through $(x, f(x) - \varepsilon)$ which supports the graph of $f(x + \alpha d)$, $\alpha \geq 0$, from below. Hence in (7.4) we compare the difference between this slope and that of the tangent at $(x, f(x))$ to the *vertical* increment $\varepsilon^{1/2}$. In (7.1) the tangents in direction d at $x + td$ and x are compared to the *horizontal* increment t. At the first glance there is no relation between these two limits. A closer analysis shows however that, thanks to convexity, they are closely related.

Theorem 7.1. *Let x and d be given. Suppose one of the limits* (7.1) *and* (7.4), *respectively, exists. Then the other one exists as well and*

$$f''(x;d) = \tfrac{1}{2}c(x;d)^2 . \qquad \square$$

The proof of the above statement is very technical.

Provided $f''(x;d)$ exists, then, because of Theorem 7.1, we may write $c(x;d)^2/4$ instead of $f''(x;d)/2$ in (7.3). Further, we will not cause a disaster if we replace $c(x;d)$ by $[f'(x;d) - f'(x;d)]/\sigma^{1/2}$ for fixed small positive σ. This leads to the modification of (7.3):

$$M_\sigma(x;d) := f'(x;d) + \frac{1}{4\sigma}[f'_\sigma(x;d) - f'(x;d)]^2 . \tag{7.5}$$

This $M_\sigma(x;\cdot)$ is a model at x which can be minimized to obtain a direction d_σ, say (x, standing for x_k, is fixed, d is the variable).

To justify this approach, we first observe that, in contrast to (7.3), the model (7.5) is well-defined even if $f''(x,d)$ does not exist. Further, all quantities in M_σ can be computed via suitable max-terms. Finally, let us show that the direction d_σ is at least as good as the direction d_ε which we use in the corresponding first order iteration (6.13). We proceed in several steps from the smooth to the nonsmooth situation.

Consider a quadratic function $f(x) := \frac{1}{2}x^\mathrm{T}Ax + b^\mathrm{T}x$ with symmetric positive definite matrix A. Straightforward computation of the inf in (6.8) gives

$$f'_\sigma(x;d) = f'(x;d) + (2\sigma d^\mathrm{T}Ad)^{1/2} , \tag{7.6}$$

hence

$$M_\sigma(x;d) = x^\mathrm{T}Ad + b^\mathrm{T}d + \tfrac{1}{2}d^\mathrm{T}Ad \equiv f(x+d) - f(x)! \tag{7.7}$$

As a result, we obtain for every $\sigma > 0$ the Newton direction:

$$d_\sigma = -A^{-1}(Ax+b) = -(\nabla^2 f(x))^{-1}\nabla f(x) . \tag{7.8}$$

Obviously there is no reason for the d_ε of Section 6.1 to be close to the Newton direction.

In case f is C^2 then for small σ our model $M_\sigma(x;d)$ is close to the second order model (3.2). Hence we can expect that, at least for small σ, the direction d_σ will be close once more to the Newton direction, whereas for small ε the first order direction d_ε will be close to the steepest descent direction.

When f is C^1 a convergence result can be proved comparable to the one for the steepest descent method and smooth f; for the general case, finally, we observe that $M_\sigma(x,0) = 0$; hence $M_\sigma(x,d_\sigma) < 0$, which obviously implies $f'(x,d_\sigma) < 0$, i.e. d_σ is at least a descent direction. It turns out that (at least if $0 \notin \partial_\sigma f(x)$!) the optimal d_σ does satisfy $f'_\sigma(x,d_\sigma) < 0$, i.e. a move in the direction d_σ guarantees a decrease of at least σ. We summarize: the above model combines the advantage of Newton's model in the region where f is smooth with a guaranteed decrease of at least σ in every step.

The above model is still waiting for an implementation. Unfortunately, the building technique of Section 6.2 essentially results in the cutting plane algorithm (5.19). Hence, some new idea is still really needed.

7.2. First order approximation of the subdifferential

Let us come back to the approximate optimality condition (6.12). As already mentioned, the point-to-set mapping $x \rightarrow \partial_\varepsilon f(x)$ varies in a Lipschitzian way with x. Even more can be said: $f'_\varepsilon(x, p)$ which, for fixed $\varepsilon > 0$ and p, varies in a Lipschitzian way with x, does have directional derivatives

$$f''_\varepsilon(x, p; d) := \lim_{t \downarrow 0} \frac{1}{t} [f'_\varepsilon(x + td, p) - f'_\varepsilon(x, p)] . \tag{7.9}$$

Hence, our motivation is as follows: since we want to solve

$$f'_\varepsilon(x, p) \geq 0 \ \ \forall p \in R^n , \quad \text{i.e. } 0 \in \partial_\varepsilon f(x) , \tag{7.10}$$

what about applying the idea of Newton's method? In other words: let us replace f'_ε in (7.10) by its first order approximation coming from (7.9), to obtain.the iteration:

$$\text{Let } d_k \text{ solve } \ f'_\varepsilon(x_k, p) + f''_\varepsilon(x_k, p; d) \geq 0 \ \ \forall p \in R^n , \tag{7.11a}$$

$$x_{k+1} := x_k + t_k d_k , \quad t_k > 0 \text{ suitable} . \tag{7.11b}$$

Several really intricate questions must be addressed before this approach can become practical. Among them are (i) how good is d_k coming from (7.11) in terms of minimizing f? (ii) when can be compute numerically a solution of (7.11)? or even: when does (7.11) have a solution? (iii) just as in (7.3) we do not have a nice expression like (1.7) for f''_ε of (7.9), and once again this will be critical for an eventual implementation. No practical answer to these questions is foreseen in the near future.

Nevertheless, the present approach poses an interesting, purely mathematical, question: just in the same way as f'_ε is related to $\partial_\varepsilon f$, can we relate the derivative in (7.9) to some derivative of the multi-valued mapping $\partial_\varepsilon f(x)$? Or, even more generally, it is possible to define a derivative of a multi-valued mapping?

Several proposals have been made in the past for such an object, each with its own motivation. For our purpose, what we need is a *first order approximation*, i.e.: let $F(t)$ be a subset of R^n depending on the parameter $t \in [0, 1]$ ($F(t)$ stands for $\partial_\varepsilon f(x + td)$), and denote by B the unit ball in R^n; we want to find another set $S(t)$, depending on the same parameter t, and such that

$$\forall \varepsilon > 0,\ \exists \delta > 0\colon\ t \in [0, \delta] \ \Rightarrow$$
$$S(t) \subset F(t) + \varepsilon t B \text{ and } F(t) \subset S(t) + \varepsilon t B\,; \tag{7.12}$$

and

the mapping S is simple, in a sense to be specified. (7.13)

We will henceforth assume that $F(t)$ is convex for each t, and therefore we will require also $S(t)$ to be convex for each t.

By analogy with the single-valued case, the simplest S would be defined via a set $F'(0) \subset R^n$ (the derivative of $F(t)$ at $t = 0$) by the equation

$$S(t) := F(0) + tF'(0)\,. \tag{7.13a}$$

Unfortunately, the existence of such an $F'(0)$ is ruled out even for mappings as simple as $F(t) := (1 - t)B$. Hence, we have to content ourselves with some more general mapping S and the next idea that comes to mind is to require

$$S(t) = (1 - t)S(0) + tS(1) \quad \forall t \in [0, 1] \tag{7.13b}$$

(assuming that $t = 1$ is in the domain of S), a property that resembles linearity, but restricted to $t \in [0, 1]$.

Even more generally, we may require only one inclusion in (7.13b) namely:

$$S(t) \supset (1 - t)S(0) + tS(1)\,, \tag{7.13c}$$

which means the more intrinsic property

S has a convex graph. (7.13d)

An advantage of (7.13d) is that it can be directly generalized to $t \in R^n$.

On the other hand, *canonicity* must also be taken into account when specifying (7.13): there should be only one S satisfying (7.12), (7.13). Indeed (7.13b) implies it, but (7.13d) does not; to recover it, a convenient supplement is maximality:

The graph of S contains the graph of all other mappings satisfying (7.12), (7.13d). (7.13d)′

Alternately, this maximal convex approximation can be defined as satisfying the property (compare (7.13a) and (7.13b)):

$$S(t) = \cap \left\{ y + t\frac{S(\tau) - y}{\tau} \,\middle|\, y \in F(0) \right\} \quad \forall \tau \in]0, t]\,. \tag{7.13e}$$

A mapping possessing the defining property (7.13e) has the following nice

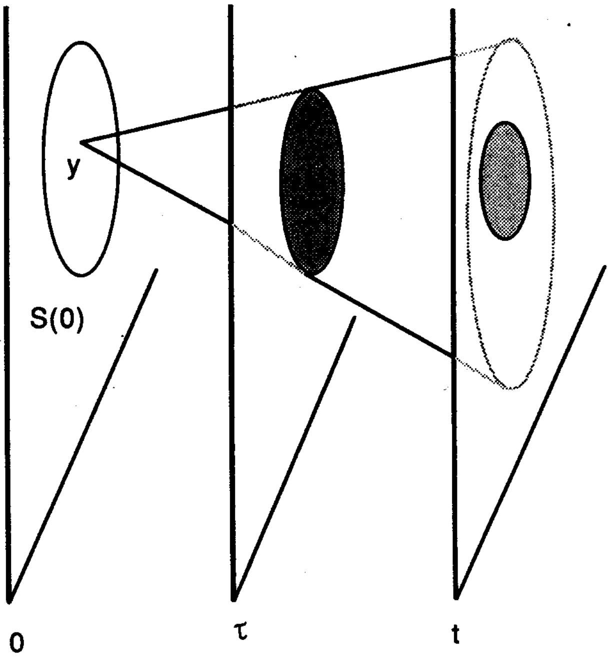

Fig. 7.1.

geometric interpretation, in 2 dimensions (see Figure 7.1): consider y as a luminous spot at $t = 0$, which illuminates the space at t; $S(\tau)$ is an opaque object in the space at τ; then there is a zone at t which is always dark, wherever y is in $F(0)$; this is just $S(t)$, and the property holds independently of the value of $\tau \leq t$.

8. Commented bibliography

As mentioned earlier, a constant reference throughout this paper is Zowe (1985), and also Lemaréchal (1980); see also the recent Kiwiel (1988).

For fairly extensive bibliographies, see Nurminskii (1982), Lemaréchal and Mifflin (1978), Kiwiel (1985).

Section 1. The reference book for all the necessary concepts from convex analysis is Rockafellar (1970). A minimal but easy to read introduction is Section 8 and the appendices of Lasdon (1970). For extensions to nonconvex situations, the basic material is in Clarke (1975, 1983). In practice, the Lipschitz assumption is too general and some additional hypotheses must be included. For convenient subclasses of Lipschitz functions see the review of Lemaréchal (1986).

Section 2. Genuinely nondifferentiable problems often appear in an infinite dimensional setting: shape optimization (Myslinski and Sokolowski (1985), Neittaanmaki and Tiba (1984)), optimal control (Outrata (1983), and more generally Clarke (1983)); operations research (Hanscom et al. (1985), Cullum et al. (1975), Chapter 9 of Grötschel et al. (1987)), Tchebychef approximation, for example in circuit design (Hald and Madsen (1981)). Another source of nonsmooth problems is in stochastic programming (Ruszczynski (1986)); we mention that, for these problems, methods of Section 5 (known there as stochastic gradient) are often most efficient (Goursat et al. (1986)).

Proposition 2.1 is due to Danskin (1966) for the case when h is maximized at a single y (i.e. when $\nabla f(x)$ exists); in its most general form, it can be found for example in Clarke (1975).

The approach in Section 2.2 is generally called Benders decomposition, see a review in Geoffrion (1972); an interesting case is when x and y are ordered in a hierarchical control system (Ruszczynski (1982)). Differential properties of the resulting f form one of the hard subjects in nonlinear analysis, with many papers devoted to it; see for example Fiacco (1984).

Section 2.3 represents an extremely important field of mathematical programming. Good treatments are in Lasdon (1970), Geoffrion (1971a,b). Even if the technique does not directly solve the problem via dual maximization (Bertsekas et al. (1983a)), duality often plays a central role – for example in integer programming (Fisher et al. (1975)). As for Section 2.4 see for example Geoffrion (1970), Hogan (1973), Silverman (1972). Here again, differential properties of $v(y)$ are a central issue, we again refer to Fiacco (1984). A combination of price- and quota-decomposition is considered in Mahey (1986). Finally, see several specific applications in operations research in Shor (1985).

Section 3. The example of non-convergence is inspired from Wolfe (1975); see also Demjanov and Malozemov (1974). An example involving only linear functions is given in Wolfe (1974). The example of Section 3.3 is due to M.J. Todd; for sophisticated use of finite differences, see Gupal (1977), p. 15 of Shor (1985) and also Theorem I.17.1 of Demjanov and Vasiliev (1986).

Section 4. The first instance of an algorithm for composite optimization was probably due to Demjanov (1968), giving what essentially is the steepest descent for (4.5) or equivalently the projected gradient method for (4.6). For penalty-type approaches, see Chapter 3 of Bertsekas (1982). For optimality conditions see Fletcher and Watson (1980), Ben Tal and Zowe (1982) (where condition (4.15) is derived); for methods based thereon, see Pschenichnyi and Danilin (1978) and Fletcher (1982).

Sometimes, methods for composite optimization can become quite sophisticated, like for minimizing the maximal eigenvalue of a varying matrix (Fletcher (1985), Overton (1988), Overton and Womersley (1988)). One would imagine that this problem has the form (4.5) but this is false; for example, the subdifferential is not a polyhedron, but an ellipsoid, see Kato (1976).

We mention also another approach for convex optimization, given in Kao and Meyer (1981); it is applicable to the *separable* case, when f is a sum of one-dimensional convex functions. The approach is original in that, rather than a lower-approximation of f based on its subgradients, it is an upper-approximation that is considered, based on convex combinations of f at iteratively computed sampling points.

Section 5. For a general description of subgradient methods, we refer to the monograph of Shor (1985). See for example Nurminskii and Zhelikhowski (1977) for various generalization.

The space dilation along the gradient was defined in Shor (1970) and its son the ellipsoid method in Judin and Nemirowski (1976). The second dilation (along the difference of gradients) dates from Shor and Zhurbenko (1971); it is much less known, though much more used in Soviet Union.

The cutting plane method is due independently to Cheney and Goldstein (1959) and Kelley (1960). The seminal works for the "school of global algorithms" were Judin and Nemirovsky (1976) where Theorems 5.6 and 5.7 were proved, while the method of centers of gravity is due to Levin (1965). Theorem 5.5 was proved in Wolfe (1970); see also p. 147 of Nemirovsky and Judin (1983); the latter book is a *must* concerning this global theory.

The idea of placing x_{k+1} at a "center" of a "safety set" exists also in the context of nonlinear programming, see Bui Trong Lieu and Huard (1966), where the terminology "*F*-distance" is introduced. The particular center of (5.21) is to be compared to those coming from projective geometry as in Karmarkar (1984), de Ghellinck and Vial (1988). Along these lines, we refer to various works such as Sonnevend (1985, 1986).

Section 6. Although the cutting plane method (5.19) itself can be considered as the very first instance of a bundle method, the ideas of Section 6 were consciously exploited in the first place by Lemaréchal (1974), and then independently by Lemaréchal (1975), Wolfe (1975). The algorithm of Section 6.3 was presented in Lemaréchal (1976) and formalized in Lemaréchal et al. (1981); its variant (6.18), defined in Lemaréchal (1978), was developed in Mifflin (1982), Kiwiel (1983); according to Kiwiel (1989), this latter variant might give definitely better results.

For the nonconvex case, Feuer (1974) realized that a mere Lipschitz assumption is not enough and adapted the bundling idea to max-functions. The minimal assumption is that of Bihain (1984) and the mathematically most satisfactory goes along semi-smoothness of Mifflin (1977).

Finally, all the above references are contained in the exhaustive monograph Kiwiel (1985), which is the best reference for a complete and detailed setting of all these methods, most refined proofs of convergence, extensions to nonconvex and constrained cases, some numerical illustrations and a fairly complete bibliography. Additional comparative results can be found in Lemaréchal (1982), Zowe (1985).

Section 7. Only few papers have been published yet, since these ideas are quite new. Concerning Section 7.1, there are only Lemaréchal and Zowe (1983) and Lemaréchal and Strodiot (1985); for Theorem 7.1 and related material, see Seeger (1986), and also Hiriart-Urruty (1986). The directional derivative (7.9) was obtained in Lemaréchal and Nurminskii (1980) and the views expressed in 7.2 come from Auslender (1982), Demjanov et al. (1986), Lemaréchal and Zowe (1988); property (7.13b) is essentially due to Gautier (1978); a good review of differentiability problems for multi-valued mappings is Penot (1984).

Bibliography

S. Agmon (1954), The relaxation method for linear inequalities, *Canadian Journal of Mathematics* **6**, 382–392.

A. Auslender (1982), On the differential properties of the support function of the subdifferential of a convex function, *Mathematical Programming* **24**(3), 257–268.

A. Ben Tal and J. Zowe (1982), Necessary and sufficient optimality conditions for a class of nonsmooth minimization problems, *Mathematical Programming* **24**(1), 70–91.

D.P. Bertsekas (1982), *Constrained Optimization and Lagrange Multiplier Methods* (Academic Press, New York).

D.P. Bertsekas, G.S. Lauer, N.R. Sandell, T.A. Posbergh (1983), Optimal short-term scheduling of large-scale power systems, *IEEE Transactions on Automatic Control* **28**(1), 1–11.

A. Bihain (1984), Optimization of upper semi differentiable functions, *Journal of Optimization Theory and Applications* **4**, 545–568.

Bui Trong Lieu and P. Huard (1966), La méthode des centres dans un espace topologique, *Numerische Mathematik* **8**, 56–67.

E.W. Cheney and A.A. Goldstein (1959), Newton's method for convex programming and Tchebycheff approximation, *Numerische Mathematik* **1**, 253–268.

F.H. Clarke (1975), Generalized gradients and applications, *Transactions of the A.M.S.* **205**, 247–262.

F.H. Clarke (1983), *Optimization and Nonsmooth Analysis* (Wiley, New York).

J. Cullum, W.E. Donath and P. Wolfe (1975), The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices, in: M.L. Balinski and P. Wolfe (eds), *Nondifferentiable Optimization*, Mathematical Programming Study 3 (North-Holland, Amsterdam) 35–55.

J.M. Danskin (1966), The theory of max–min with applications, *SIAM Journal on Applied Mathematics* **14**(4), 641–655.

V.F. Demjanov (1968), Algorithms for some minimax problems, *Journal of Computer and System Science* **2**, 342–380.

V.F. Demjanov, C. Lemaréchal and J. Zowe (1986), Approximation to a set-valued mapping I: A proposal, *Applied Mathematics and Optimization* **14**(3), 203–214.

V.F. Demjanov and V.N. Malozemov (1974), *Introduction to Minimax* (Wiley, New York).

V.F. Demjanov and L.V. Vasiliev (1985), *Nondifferentiable Optimization* (Optimization Software, Inc./Springer-Verlag, Berlin).

A. Feuer (1974), An implementable mathematical programming algorithm for admissible fundamental functions, Ph.D. Thesis, Dept of Mathematics, Columbia Univ.

A.V. Fiacco (ed.) (1984), *Sensitivity, Stability and Parametric Analysis*. Mathematical Programming Study 21 (North-Holland, Amsterdam).

M.L. Fischer, W.D. Northup and J.F. Shapiro (1975), Using duality to solve discrete optimization problems: theory and computational experience, in: M.L. Balinski and P. Wolfe (eds.), *Nondifferentiable Optimization*, Mathematical Programming Study 3 (North-Holland, Amsterdam) 56–94.

R. Fletcher (1982), Second order corrections for nondifferentiable optimization, in: G.A. Watson (ed.), *Numerical Analysis*, Lectures Notes in Mathematics 912, (Springer-Verlag, Berlin) 85–114.

R. Fletcher (1985), Semi-definite matrix constraints in optimization, *SIAM Journal on Control and Optimization* **23**(4), 493–513.

R. Fletcher and G.A. Watson (1980), First and second order conditions for a class of nondifferentiable optimization problems. *Mathematical Programming* **18**(3), 291–307.

S. Gautier (1978), Différentiabilité des multi-applications, Working Paper, Dept. of Mathematics, Univ. of Pau.

A.M. Geoffrion (1970), Primal resource-directive approaches for optimizing nonlinear decomposable systems, *Operations Research* **18**(3), 375–403.

A.M. Geoffrion (1971a), Elements of large scale programming, *Management Science* **16**(11), 652–675.

A.M. Geoffrion (1971b), Duality in nonlinear programming: A simplified application-oriented development, *SIAM Review* **13**(11), 1–37.

A.M. Geoffrion (1972), Generalized Benders decomposition, *Journal of Optimization Theory and Applications* **10**(4), 237–260.

G. de Ghellinck and J.P. Vial (1986), A polynomial Newton method for linear programming, *Algorithmica* **1**(4), 425–453.

M. Goursat, J.P. Quadrat and M. Viot (1986), Stochastic gradient methods for optimizing electrical transportation netwotds, in: V.I. Arkin, A. Shiryaev, R. Wets (eds.), *Stochastic Optimization*. Lecture Notes in Control and Information Sciences 81 (Springer-Verlag, Berlin) 373–387.

M. Grötschel, L. Lovasz and A. Schrijver (1987), *The Ellipsoid Method and Combinatorial Optimization* (Springer-Verlag, Berlin).

A.M. Gupal (1977), A method for the minimization of almost-differentiable functions, *Cybernetics* **13**(1), 115–117.

J. Hald and K. Madsen (1981), Combined LP and quasi-Newton methods for minimax optimization, *Mathematical Programming* **20**(1), 49–62.

M.A. Hanscom, V.H. Nguyen and J.J. Strodiot (1985), A reduced subgradient algorithm for network problems with convex nondifferentiable costs, in: V.F. Demjanov and D. Pallaschke (eds.), *Nondifferentiable Optimization: Motivations and Applications*, Lecture Notes in Economics and Mathematical Systems 255 (Springer-Verlag, Berlin) 318–322.

J.B. Hiriart-Urruty (1986), A new set-valued second order derivative for convex functions, in: J.B. Hiriart-Urruty (ed.), *Fermat Days 85: Mathematics for Optimization* (North-Holland, Amsterdam) 157–182.

W. Hogan (1973), Directional derivatives for extremal-valued functions with applications to the completely convex case, *Operations Research* **20**(1), 188–209.

D.B. Judin and A.S. Nemirovskii (1976), Estimation of the informational complexity of mathematical programming problems, *Matekon* **13**, 2–45.

C.Y. Kao and R.R. Meyer (1981), Secant approximation methods for convex optimization, in: H. König, B. Korte, K. Ritter (eds.), *Mathematical Programming Study* 14 (North-Holland, Amsterdam) 143–162.

N. Karmarkar (1984), A new polynomial time algorithm for linear programming, *Combinatorica* **4**(4), 373–395.

T. Kato (1976), *Perturbation Theory for linear operators* (Springer-Verlag, Berlin).

J.E. Kelley (1960). The cutting plane method for solving convex programs, *Journal of the SIAM* 8, 703–712.

K.C. Kiwiel (1983), An aggregate subgradient method for nonsmooth convex minimization, *Mathematical Programming* **27**(3), 320–341.

K.C. Kiwiel (1985), *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics 1133 (Springer-Verlag, Berlin).

K.C. Kiwiel (1988), A survey of bundle methods for nondifferentiable optimization, *Proceedings XIII International Symposium on Mathematical Programming* (Tokyo).

K.C. Kiwiel (1989), Proximity control in bundle methods for convex nondifferentiable minimization, *Mathematical Programming* (to appear).

L.S. Lasdon (1970), *Optimization Methods for Large Scale Problems* (MacMillan, New York).

C. Lemaréchal (1974), An algorithm for minimizing convex functions, in: J.L. Rosenfeld (ed.), *Proceedings IFIP'74 Congress* (North-Holland, Amsterdam) 552–556.

C. Lemaréchal (1975), An extension of Davidon methods to nondifferentiable problems, in: M.L. Balinski and P. Wolfe (eds.), *Mathematical Programming Study* 3 (North-Holland, Amsterdam) 95–109.

C. Lemaréchal (1976), Combining Kelley's and conjugate gradient methods, Abstracts, IX International Symposium on Mathematical Programming (Budapest), 158–159.

C. Lemaréchal (1978), Nonsmooth optimization and descent methods, Report RR 784, IIASA, 2361 Laxenburg (Austria).

C. Lemaréchal (1980), Nondifferentiable optimization, in: L.C.W. Dixon, E. Spedicato, G.P. Szegö (eds.), *Nonlinear Optimization* (Birkhäuser, Basel) 149–199.

C. Lemaréchal (1982), Numerical experiments in nonsmooth optimization, in: E. A. Nurminski (ed.), *Progress in nonsmooth optimization*, Proceedings CP 82.58, IIASA, 2361 Laxenburg (Austria), 61–84.

C. Lemaréchal (1986), Basic theory in nondifferentiable optimization, *Optimization* **17**(6) 827–858.

C. Lemaréchal and R. Mifflin (eds.) (1978), *Nonsmooth Optimization* (Pergamon Press, Oxford).

C. Lemaréchal and E.A. Nurminski (1980), Sur la différentiabilité de la fonction d'appui du sous-différentiel approché, *Comptes Rendus Académie des Sciences Paris* **290**(18), 855–858.

C. Lemaréchal and J.J. Strodiot (1985), Bundle methods, cutting-plane algorithms and σ-Newton directions, in: V.F. Demjanov and D. Pallaschke (eds.), *Nondifferentiable Optimization*, Lecture Notes in Economics and Mathematical Systems 255 (Springer-Verlag, Berlin) 25–33.

C. Lemaréchal, J.S. Strodiot and A. Bihain (1981), On a bundle algorithm for nonsmooth optimization, in: O.L. Mangasarian, G.L. Meyer, S.M. Robinson (eds.), *Nonlinear Programming 4* (Academic Press, New York) 245–282.

C. Lemaréchal and J. Zowe (1983), Some remarks on the construction of higher order algorithms for convex optimization, *Applied Mathematics and Optimization* **10**(1), 51–68.

C. Lemaréchal and J. Zowe (1987), Approximation to a set valued mapping II: Existence, uniqueness, characterization, Schwerpunktprogramm der Deutschen Forschungsgemeinschaft "Anwendungsbezogene Optimierung und Steuerung", Report No. 5.

A.Y. Levin (1965), On an algorithm for minimizing a convex function, *Soviet Mathematics Doklady* **6**(1), 286–290.

P. Mahey (1986), Méthodes de décomposition et décentralisation en programmation linéaire, *RAIRO Rech. Opér.* **20**(4), 287–306.

R. Mifflin (1977), Semi-smooth and semi-convex functions in constrained optimization, *SIAM Journal on Control and Optimization* **15**(6), 959–972.

R. Mifflin (1982), A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization, in: D.C. Sorensen and R.J.B. Wets (eds.), *Mathematical Programming Study* 17, (North-Holland, Amsterdam) 77–90.

T. Motzkin and I. Schönberg (1954), The relaxation method for linear inequalities, *Canadian Journal of Mathematics* **6**, 393–404.

A. Myslinski and J. Sokolowski (1985), Nondifferentiable optimization problems for elliptic systems, *SIAM Journal on Control and Optimization* **23**(4), 632–648.

P. Neittaanmaki and D. Tiba (1984), On the finite element approximation of the boundary control for two-phase Stefan problems, in: A. Bensoussan J.L. Lions (eds.), *Analysis and Optimization of Systems*, Lecture Notes in Control and Information Sciences 62 (Springer-Verlag, Berlin) 356–370.

A.S. Nemirovsky and D.B. Yudin (1983), *Problem complexity and method efficiency in optimization* (Wiley, New York).

E.A. Nurminski (ed.) (1982), *Progress in Nondifferentiable Optimization*, Publication CP-82-58, IIASA, 2361 Laxenburg, Austria.

E.A. Nurminski and A.A. Zhelikhowskii (1977), ε-quasi-gradient method for solving nonsmooth extremal problems, *Cybernetics* **13**(1), 109–114.

J.V. Outrata (1983), On a class of nonsmooth optimal control problems, *Applied Mathematics and Optimization* **10**(4), 287–306.

M.L. Overton (1988), On minimizing the maximum eigenvalue of a symmetric matrix, *SIAM Journal on Matrix Analysis and Applications* **9**, 256–268.

M.L. Overton and R.S. Womersley (1988), On minimizing the spectral radius of a nonsymmetric matrix function - Optimality conditions and duality theory, *SIAM Journal on Matrix Analysis and Applications* **9**, 473–498.

J.P. Penot (1984), Differentiability of relations and differential stability of perturbed optimization problems, *SIAM Journal on Control and Optimization* **22**(4), 529–551.

B.T. Poljak (1977), Subgradient methods: A survey of Soviet research, in: C. Lemaréchal and R. Mifflin (eds.) *Nonsmooth Optimization* (Pergamon Press, Oxford) 5–30.

B.N. Pschenichny and Y.M. Danilin (1978), *Numerical Methods for Extremal Problems* (Mir, Moscow).

R.T. Rockafellar (1970), *Convex Analysis* (Princeton University Press, Princeton).

A. Ruszczynski (1982), Nondifferentiable functions in hierarchical control problems, in: E.A. Nurminski (ed.), *Progress in Nondifferentiable Optimization*, Publication CP-82-58, IIASA, 2361 Laxenburg, Austria, 145–172.

A. Ruszczynski (1986), A linearization method for nonsmooth stochastic programming problems, *Mathematics of Operations Research* **12**(1), 32–49.

A. Seeger (1986), Analyse du second ordre de problèmes non différentiables, Ph.D. Thesis, Dept. of Mathematics, Univ. of Toulouse.

N.Z. Shor (1970), Utilization of the operation of space dilatation in the minimization of convex functions, *Cybernetics* **6**(1), 7–15.

N.Z. Shor (1985), *Minimization Methods for Nondifferentiable Functions* (Springer-Verlag, Berlin).

N.Z. Shor and N.G. Zhurbenko (1971), A minimization method using the operation of extension of the space in the direction of the difference of two successive gradients, *Cybernetics* **7**(3), 450–459.

G.J. Silverman (1972), Primal decomposition of mathematical programs by resource allocation. I Basic theory and a direction finding procedure. II Computational algorithm with an application to the modular design programming, *Operation Research* **20**(1), 58–93.

G. Sonnevend (1985), A modified ellipsoid method for the minimization of convex functions with superlinear convergence [finite termination] for well-conditioned C^3 smooth [piecewise linear] functions, in: V.F. Demjanov and D. Pallaschke (eds.), *Nondifferentiable Optimization*. Lecture Notes in Economics and Mathematical Systems 255 (Springer-Verlag, Berlin) 264–277.

G. Sonnevend (1986), A new method for solving a set of linear (convex) inequalities and its applications, in: B. Martos (ed.), *Proceedings 5th IFAC-IFORS Symposium on Dynamic Modelling* (Pergamon Press, Oxford).

P. Wolfe (1970), Convergence theory in nonlinear programming, in: J. Abadie (ed.), *Integer and Nonlinear Programming* (North-Holland, Amsterdam) 1–36.

P. Wolfe (1974), A method of conjugate subgradients for minimizing nondifferentiable functions, Proceedings, 12th Annual Allerton Conference on Circuit and System Theory, University of Illinois at Urbana, Champaign 8–15.

P. Wolfe (1975), A method of conjugate subgradients for minimizing nondifferentiable functions, in: M.L. Balinski and P. Wolfe (eds.), *Nondifferentiable Optimization*, Mathematical Programming Study 3 (North-Holland, Amsterdam) 145–173.

J. Zowe (1985), Nondifferentiable optimization, in: K. Schittkowski (ed.), *Computational Mathematical Programming* (Springer-Verlag, Berlin) 323–356.