SIAM J. OPTIMIZATION
Vol. 2, No. 1, pp. 121–152, February 1992

008

A VERSION OF THE BUNDLE IDEA FOR MINIMIZING A NONSMOOTH FUNCTION: CONCEPTUAL IDEA, CONVERGENCE ANALYSIS, NUMERICAL RESULTS*

HELGA SCHRAMM† AND JOCHEM ZOWE†

Abstract. During recent years various proposals for the minimization of a nonsmooth functional have been made. Amongst these, the bundle concept turned out to be an especially fruitful idea. Based on this concept, a number of authors have developed codes that can successfully deal with nonsmooth problems. The aim of the paper is to show that, by adding some features of the trust region philosophy to the bundle concept, the end result is a distinguished member of the bundle family with a more stable behaviour than some other bundle versions. The reliability and efficiency of this code is demonstrated on the standard academic test examples and on some real-life problems.

Key words. nondifferentiable optimization, bundle methods

AMS(MOS) subject classifications. 90C30, 65K05

1. Introduction and exposition of conceptual ideas.

This paper deals with the minimization of a *nonsmooth* functional (i.e., $f \notin C^1$)

$$\text{minimize } f(x) \text{ where } f : \mathbb{R}^n \to \mathbb{R}. \tag{1.1}$$

Additional constraints in (1.1) do not cause difficulties—at least in theory; they can be added to f as (nonsmooth) exact penalty terms. As usual in *nonsmooth optimization* (NSO), we require throughout that

$$f \text{ is locally Lipschitzian.} \tag{1.2}$$

For such f the *subdifferential* of f at x,

$$\partial f(x) := \operatorname{conv}\{g \in \mathbb{R}^n \mid g = \lim_{i\to\infty} \nabla f(x_i),\ x_i \to x, \nabla f(x_i) \text{ exists},\ \nabla f(x_i) \text{ converges}\}, \tag{1.3}$$

is a well-defined, nonempty, convex, and compact subset of $\mathbb{R}^n$; this and other standard facts from convex analysis and NSO can be found, e.g., in the textbooks by Clarke [4] and Rockafellar [35]. The elements of $\partial f(x)$ are called *subgradients* of f at x. Quite naturally, these subgradients serve in NSO as substitute for the gradients. Hence, parallel to what is standard in *smooth optimization*, we require in the following that we dispose of a subroutine that

$$\text{computes } f(x) \text{ and one (arbitrary) } g \in \partial f(x) \text{ for given } x. \tag{1.4}$$

This seems to be a modest (and minimal) requirement. Sections 4.2–4.4, however, will show that in many real-life situations, the computation of only one $g \in \partial f(x)$ is all but easy and is the time-consuming job per iteration.

* Received by the editors March 26, 1990; accepted for publication (in revised form) December 18, 1990. This research has been supported by Deutsche Forschungsgemeinschaft, Forschungsschwerpunkt "Anwendungsbezogene Optimierung und Steuerung."

† Mathematisches Institut, Universität Bayreuth, Postfach 101251, 8580 Bayreuth, Germany.

1.1. Subgradient methods. Apart from the cutting plane method (see below), the first methods which could deal with (1.1) under assumption (1.4) and for convex f were the *Russian subgradient methods* (also called *Kiev methods*); see, e.g., Ermoliev [9], Poljak [34], and Shor [39]. At iterate x_k one makes a step along a negative subgradient with some off-line chosen steplength

$$x_{k+1} := x_k + \lambda_k d_k \quad \text{where } d_k := -g_k/\|g_k\| \quad \text{with } g_k \in \partial f(x_k)\,. \tag{1.5}$$

It can be shown that, under rather suggestive assumptions and for $\lambda_k \downarrow 0$ and $\sum_{k=1}^{\infty} \lambda_k = \infty$, the x_k from (1.5) converge to an optimal point. The simple structure of these subgradient methods still makes them widely used, although they suffer from some serious drawbacks: The methods do not guarantee a descent at each step, they lack an implementable stopping criterion, and the convergence speed is extremely poor (less than linear). The last disadvantage can be partly overcome by premultiplying d_k in (1.5) with some variable metric matrix H_k, which is updated in a simple way at each iteration. Linear convergence in the function values can be established for a member of this class (see, e.g., Shor [39]); the additional H_k, however, makes the method very cumbersome for large n.

1.2. Bundle concept. Lemaréchal [23] and Wolfe [41] initiated a giant stride forward in NSO by the *bundle concept*, which can handle convex and nonconvex f. Since the motivating ideas come from the convex situation, we assume a convex f throughout this motivating section.

All bundle methods carry two distinctive features (for some other views in this section, see Lemaréchal [24] and Zowe [43]):

(i) They make use at the iterate x_k of the *bundle* of information $(f(x_k), g_k)$, $(f(x_{k-1}), g_{k-1}), \cdots$ collected so far to build up a model of f;

(ii) If, due to the kinky structure of f, this model is not yet an adequate one, then they mobilize even more subgradient information close to x_k.

Recipe (i) leads in a natural way to the *cutting plane* (CP) *approximation* of f at x_k:

$$\max_{1 \le i \le k} \{g_i^T (x - x_i) + f(x_i)\}\,. \tag{1.6}$$

Equation (1.6) is a piecewise linear approximation of the convex f from below, which coincides with f at all x_i. For short, we put $d := x - x_k$ in (1.6) and use the notation

$$f_{\mathrm{CP}}(x_k; d) := \max_{1 \le i \le k} \{g_i^T d + g_i^T (x_k - x_i) + f(x_i)\} \quad \text{for } d \in \mathbb{R}^n\,. \tag{1.7}$$

Obviously there is no reason to trust this substitute for f far away from x_k. Therefore a stabilizing term $(1/2t_k)d^T d$ with positive t_k is added in (1.7), when minimizing this CP-model of f. If f_{CP} models f well enough close to x_k, then the minimizer d_k of

$$f_{\mathrm{CP}}(x_k; d) + \frac{1}{2t_k} d^T d$$

is a descent direction for f and a linesearch along $x_k + \lambda d_k$ for $\lambda \ge 0$ provides some x_{k+1} with $f(x_{k+1}) < f(x_k)$. For a nonsmooth f it may happen, however, that f_{CP} is such a poor approximation of f that d_k is not a descent direction for f (or that the linesearch only leads to a marginal decrease in f); think, e.g., of $f(x) = |x|$, $x_i < 0$ for $i = 1, \cdots, k$ and x_k close to the kink 0. Here strategy (ii) comes up: Obviously f_{CP} does not copy f on the halfline $x_k + \lambda d_k$, $\lambda \ge 0$; to master this lack of

information one stays at x_k and enriches the model by including one more subgradient from $\partial f(x_k + \lambda d_k)$ for small $\lambda > 0$. Omitting all details, we obtain the following.

(1.8) **Iteration** $x_k \to x_{k+1}$**:**

(1) Compute $d_k := d(t_k) := \arg\min\{f_{\mathrm{CP}}(x_k; d) + (\frac{1}{2t_k})d^T d \mid d \in \mathbb{R}^n\}$.

(2) Perform a linesearch for f along $x_k + \lambda d_k$, $\lambda \geq 0$.

(a) If the linesearch leads to a "sufficient decrease" in f, then make a **Serious Step**: Put $x_{k+1} := x_k + \lambda_k d_k$ with $\lambda_k \in \arg\min_{\lambda > 0} f(x_k + \lambda d_k)$ and compute $g_{k+1} \in \partial f(x_{k+1})$.

(b) If the linesearch yields only an "insufficient decrease," then make a **Null Step**: Put $x_{k+1} := x_k$ and compute $g_{k+1} \in \partial f(x_k + \lambda d_k)$ for suitable small $\lambda > 0$.

Unlike the subgradient approach, the above iteration guarantees a decrease for each (Serious) Step. Further, one disposes of an implementable stopping criterion: x_k is "optimal" as soon as d_k in (1) is "close" to 0. And, since the linesearch adjusts the steplength λ_k to the chosen d_k, one has a considerably faster convergence speed. All this can be made precise and a detailed convergence analysis exists for convex and nonconvex f; see Lemaréchal, Strodiot, and Bihain [26]; Mifflin [31]; or the monograph by Kiwiel [18].

The above concept has been implemented by a number of authors. We mention in particular the advanced and sophisticated Fortran code M1FC1 by Lemaréchal [27], which is widely used in NSO. Numerous test runs proved the efficiency of this code; see also §4 below. Needless to say, M1FC1 is only "work of man!" The reader who is familiar with M1FC1 in applications will agree that the code suffers from two weak points: First, the success of M1FC1 depends in a delicate way on the parameter t_k in step (1) of (1.8) (actually some "dual" parameter ε_k is used in M1FC1); a bad guess for t_k (respectively, ε_k) leads to a "bad" search direction d_k and M1FC1 breaks down with linesearch difficulties. Second, for $f \in C^1$ and $t_k \to 0$, (1.8) reduces to the *steepest descent method*, which is only linearly convergent. Numerical experiments confirm this first-order behaviour of M1FC1. We will discuss how one can bypass the first shortcoming in practice; further, it will become obvious how to deal, in principle, with the second problem and how to reach faster convergence.

1.3. Bundle trust region concept. We start with a simple observation: With d_k from step (1) of iteration (1.8) and $\rho_k := \frac{1}{2} d_k^T d_k$, the minimization in (1.8)(1) becomes "equivalent" to

$$\text{compute } d(\rho_k) := \arg\min\{f_{\mathrm{CP}}(x_k; d) \mid \tfrac{1}{2} d^T d \leq \rho_k\}. \tag{1.9}$$

This follows by a comparison of the Kuhn–Tucker conditions for the two problems. A closer inspection shows that there is even a monotone correspondence between t_k and ρ_k. Now we replace (1.8)(1) by (1.9). It then becomes obvious how to bypass the first difficulty discussed above. Instead of working with some a priori and more or less randomly chosen ρ_k (respectively, t_k), we follow the *trust region* philosophy: We decrease and/or increase ρ_k in a systematic way (*trust region part*) and improve f_{CP} by Null Steps (*bundle part*), until we reach some f_{CP} together with a ρ_k-ball, on which we can *trust* this model, i.e., the d_k from (1.9) leads to a substantial decrease in f. The advantage of this procedure is twofold: It suggests a way to choose ρ_k, and it releases us at the same time from the need for a linesearch. Obviously we can apply

just the same strategy in (1.8) and tune the t_k. The reason for working with (1.8) is purely numerical in nature. We will see that the minimization of (1.8) leads to a quadratic programming problem with a lot of reliable software (e.g., [19]). This is not true for (1.9) because of the quadratic constraint. In schematic terms, we obtain

(1.10) **Iteration** $x_k \to x_{k+1}$**:**

(1) Compute $d_k := d(t_k) := \arg\min\{f_{\mathrm{CP}}(x_k; d) + \frac{1}{2t_k} d^T d \mid d \in \mathbb{R}^n\}$.

(2) If $f(x_k + d_k)$ is "sufficiently smaller" than $f(x_k)$, then either

(a) enlarge t_k and go back to (1), or

(b) make a **Serious Step**: Put $x_{k+1} := x_k + d_k$, compute $g_{k+1} \in \partial f(x_{k+1})$.

If $f(x_k + d_k)$ is "not sufficiently smaller" than $f(x_k)$, then either

(c) reduce t_k and go back to (1), or

(d) make a **Null Step**: Put $x_{k+1} := x_k$, compute $g_{k+1} \in \partial f(x_k + d_k)$.

How to solve the alternatives (a)–(b) and (c)–(d) will be seen in the precise statement of the algorithm.

Preliminary versions of the above variant of the bundle family have been presented in [38] and [44]. In these versions the t_k were reduced in (2)(c) only as long as they stayed above some fixed positive lower bound $\underline{t}$ (otherwise one had to make a Null Step). This restrictive assumption for the inner iteration (2)(c) can be skipped now by introducing a modified Null Step criterion (see also [21]). The above variant has been implemented by us under the name BT (= "implicit" **b**undle **t**rust region) algorithm (see the remark before Theorem 2.3). Extensive testing (in particular on some real-life problems, which are known as "tough nuts") proved the code to be efficient and reliable so far. We want to convince the reader of this claim and encourage him to work with our code and other bundle implementations.

Let us briefly return to the second drawback of the existing bundle implementations, namely the linear (hence slow) convergence. Obviously, the trust region approach could also help with this difficulty by tuning the bilinear form $d^T d$ in step (1) of (1.10) to account for the compiled knowledge about the level sets of f. There is a whole series of recent papers that address this challenging item and try to gain control of such curvature (hence second-order) information by using ideas from the ellipsoid method; see, e.g., Goffin [11]; Goffin, Haurie, and Vial [13]; Sonnevend and Stoer [40]; Kiwiel [20]. Only some first attempts for implementing these concepts have been made. Some more abstract approaches to second-order ideas in NSO are reviewed in [24].

We mention that our work has benefitted greatly from cooperation with Lemaréchal and from the work of Kiwiel (in particular, [18]). In a recent paper [21], Kiwiel proposed a bundle variant, which is close to our BT-iteration. The difference is that Kiwiel does not adapt the t in some inner iteration as we do in steps (2)(a) and (2)(c), because he does not work with the trust region philosophy. His t is updated *after* having made a Serious Step or a Null Step.

The paper is organized as follows. Since the motivation and the key arguments are based on convexity, we treat the convex case in detail in §2, i.e., we will specify iteration (1.10) together with the overall algorithm and present the convergence analysis for convex f. Section 3 discusses the necessary modifications for nonconvex f and states the convergence results without proofs; the detailed proofs can be found in Schramm [37]. Section 4 will verify our claim that BT behaves well in practice.

Some remarks on the notation: $\|\cdot\|$ denotes the Euclidean norm. The subscript k always refers to the sequence of iterates $x_1, x_2, \cdots$, whereas the superscript j will be used in the inner iteration, which leads from x_k to x_{k+1}. If J is a set of indices, then $|J|$ denotes its cardinality. Further, we put

$$\Lambda(n) := \left\{ \lambda \in \mathbb{R}^n \mid \lambda_i \geq 0,\ 1 \leq i \leq n, \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

2. BT-algorithm: The convex case. We assume throughout this section that f is convex. Then the elements of the subdifferential can be characterized by an inequality:

$$g \in \partial f(x) \iff g^T(y-x) \leq f(y) - f(x) \quad \text{for all } y \in \mathbb{R}^n. \tag{2.1}$$

This *subgradient inequality* plays a crucial role for the conceptual ideas and in the convergence analysis. For later use we add a continuity result for the set-valued map $x \to \partial f(x)$:

(2.2) the map $x \to \partial f(x)$ is locally bounded and upper semicontinuous.

Further, let us mention that a convex $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitzian, i.e., our general continuity assumption (1.1) holds.

2.1. The cutting plane model. At the iterate x_k we have at our disposal the sequence $x_1, x_2, \cdots, x_k$ and a collection of auxiliary points y_i together with subgradients $g_i \in \partial f(y_i)$ for $i \in J_k$; here J_k is some nonempty set of indices. On first reading, the reader may think of J_k as a subset of $\{1, \cdots, k\}$ and assume $y_i = x_i$. This bundle of information leads to the *cutting plane model* $\max_{i \in J_k}\{g_i^T(x - y_i) + f(y_i)\}$ of f. With the *linearization errors*

$$\alpha_{k,i} := \alpha(x_k, y_i) := f(x_k) - (f(y_i) + g_i^T(x_k - y_i)) \tag{2.3}$$

and the new variable $d := x - x_k$, we can write this in a condensed form

$$\max_{i \in J_k}\{g_i^T d - \alpha_{k,i}\} + f(x_k) \quad \text{for } d \in \mathbb{R}^n.$$

For convenience, let us skip the constant $f(x_k)$ and put

$$f_{\mathrm{CP}}(x_k; d) := \max_{i \in J_k}\{g_i^T d - \alpha_{k,i}\} \quad \text{for } d \in \mathbb{R}^n. \tag{2.4}$$

Step (1) from iteration (1.10) becomes, for *suitable* t (which still has to be chosen appropriately!):

$$\text{compute } d := d(t) = \arg\min \left\{ f_{\mathrm{CP}}(x_k; d) + \frac{1}{2t}\|d\|^2 \mid d \in \mathbb{R}^n \right\}. \tag{2.5}$$

This can equivalently be written as a quadratic programming problem in $\mathbb{R}^1 \times \mathbb{R}^n$:

$$\begin{aligned} &\text{compute } (v, d) := (v(t), d(t)) \\ &\qquad = \arg\min \left\{ v + \tfrac{1}{2t}\|d\|^2 \mid v \geq g_i^T d - \alpha_{k,i} \text{ for } i \in J_k \right\}. \end{aligned} \tag{2.6}$$

Problem (2.5) is a strictly convex problem with a unique minimizer $d(t)$; the same holds for (2.6), of course. From the Kuhn–Tucker conditions for (2.6), one easily obtains a representation for $d(t)$ and $v(t)$.

LEMMA 2.1. *For the solution* $(v(t), d(t))$ *of* (2.6) *there exists* $\lambda(t) \in \Lambda(|J_k|)$ *such that*

$$\lambda_i(t)(-v(t) + g_i^T d(t) - \alpha_{k,i}) = 0 \quad \text{for } i \in J_k\,, \tag{2.7}$$

$$d(t) = -t \sum_{i \in J_k} \lambda_i(t) g_i\,, \tag{2.8}$$

$$v(t) = -t \left\| \sum_{i \in J_k} \lambda_i(t) g_i \right\|^2 - \sum_{i \in J_k} \lambda_i(t) \alpha_{k,i} = -\frac{1}{t} \|d(t)\|^2 - \sum_{i \in J_k} \lambda_i(t) \alpha_{k,i}\,. \tag{2.9}$$

Since (2.6) is a convex problem with linear constraints, the Kuhn–Tucker conditions (i.e., (2.7)–(2.9)) are also sufficient for optimality of a feasible x.

Thanks to convexity, all $\alpha_{k,i}$ are nonnegative (a consequence of (2.1)),

$$\alpha_{k,i} \geq 0 \quad \text{for } i \in J_k. \tag{2.10}$$

Now add $\alpha_{k,i} - [f(x_k) - f(y_i) - g_i^T(x_k - y_i)] = 0$ to the subgradient inequality

$$g_i^T(x - y_i) \leq f(x) - f(y_i);$$

one obtains, after simple reordering,

$$g_i^T(x - x_k) \leq f(x) - f(x_k) + \alpha_{k,i} \quad \text{for all } x \in \mathbb{R}^n \quad \text{and} \quad i \in J_k\,, \tag{2.11}$$

i.e., $\alpha_{k,i}$ "measures" how good $g_i \in \partial f(y_i)$ satisfies the subgradient inequality at the point x_k. The $\alpha_{k,i}$ take care that the influence of g_i in (2.6) and (2.16) below will be greater the smaller the weight $\alpha_{k,i}$ is.

Now fix some $\lambda \in \Lambda(|J_k|)$, multiply (2.11) by λ_i, and sum up over i. We obtain the useful formula, which holds with arbitrary $\lambda \in \Lambda(|J_k|)$:

$$\left(\sum_{i \in J_k} \lambda_i g_i \right)^T (x - x_k) \leq f(x) - f(x_k) + \sum_{i \in J_k} \lambda_i \alpha_{k,i} \quad \text{for all } x \in \mathbb{R}^n\,. \tag{2.12}$$

Inequality (2.12) can be interpreted similarly to (2.11) above.

As a direct conclusion from (2.9) and (2.10) we note:

$$v(t) \leq 0 \quad \text{for the optimal } v(t) \text{ from (2.6).} \tag{2.13}$$

As expected, $v(t) = 0$ characterizes optimality of x_k. This follows immediately from our next result, if we put there $\varepsilon = 0$ and use (2.9). The lemma itself is an immediate consequence of inequality (2.12).

LEMMA 2.2. *Suppose there exists* $\lambda \in \Lambda(|J_k|)$ *with*

$$\left\| \sum_{i \in J_k} \lambda_i g_i \right\| \leq \varepsilon \quad \textit{and} \quad \sum_{i \in J_k} \lambda_i \alpha_{k,i} \leq \varepsilon\,. \tag{2.14}$$

Then x_k *is* ε*-optimal, i.e.,*

$$f(x_k) \leq f(x) + \varepsilon \|x - x_k\| + \varepsilon \quad \textit{for all } x \in \mathbb{R}^n\,.$$

For later use we add a continuity result on $(v(t), d(t))$, which follows easily from the strict convexity of the objective function in (2.5):

(2.15) The solution $(v(t), d(t))$ of (2.6) depends continuously on $t \in (0, \infty)$.

Due to the simple structure of (2.6), the last statement can be strengthened substantially. We add without proof (a detailed treatment is given in [37]):

- There exists a finite sequence $0 = t^0 < t^1 < \cdots < t^m = \infty$ and $a^i, b^i \in \mathbb{R}^n$, such that $d(t) = a^i + tb^i$ for $t \in (t^{i-1}, t^i]$ and $i = 1, 2, \cdots, m$;
- $a^1 = 0$ and $b^1 =$ projection of the origin onto conv $\{g_i \mid i \in J_k$ and $\alpha_{k,i} = 0\}$;
- There exists a CP-solution d_{CP} (i.e., d_{CP} minimizes $f_{\mathrm{CP}}(x_k; \cdot)$) if and only if $a^m = d_{CP}$ and $b^m = 0$.

Remark. For an efficient implementation of (2.6), two devices become important.

(a) The index set J_k (i.e., the number of subgradients carried along) should be kept at reasonable size as $k \to \infty$. Hence from time to time we clean up the bundle. The convergence analysis requires $|J_k| \geq 3$ together with a certain *reset strategy*.

(b) Problem (2.6) is a quadratic programming problem in $1+n$ variables and $|J_k|$ linear constraints. Since, typically, $|J_k|$ will be much smaller than the dimension n, we replace (2.6) by its *dual* in $|J_k|$ variables and $|J_k| + 1$ constraints:

$$\min \left\{ \frac{1}{2} \left\| \sum_{i \in J_k} \lambda_i g_i \right\|^2 + \frac{1}{t} \sum_{i \in J_k} \lambda_i \alpha_{k,i} \mid \lambda \in \Lambda(|J_k|) \right\}. \tag{2.16}$$

Some standard duality arguments show that the solutions λ of (2.16) and the $\lambda(t)$ from Lemma 2.1 correspond to each other.

In the next section we make clear how to find an appropriate t for (2.6). Section 2.3 summarizes the overall algorithm and §2.4 presents the convergence analysis.

2.2. Inner iteration $x_k \to x_{k+1}$. We fix an upper bound T for t, parameters $0 < m_1 < m_2 < 1$, $0 < m_3 < 1$, some small $\nu > 0$, and a stopping parameter $\varepsilon \geq 0$. Suppose we are at the iterate x_k and let J_k, y_i, $g_i \in \partial f(y_i)$ and $\alpha_{k,i}$ be as discussed above. Then we specialize (1.10) as follows. Here the superscript j is the running index; the subscript k is kept fixed. The stopping rule in step (1) is based on (2.8), (2.9), and Lemma 2.2. Finally, the decisive criteria **SS** and **NS** will be specified below.

(2.17) **Inner iteration $x_k \to x_{k+1}$:**

(0) Choose $t^1 := t_{k-1}$. Set $l^1 := 0$, $u^1 := T$, and $j := 1$.

(1) Compute the solution $(v^j, d^j) = (v(t^j), d(t^j))$ of (2.6). If $(1/t^j)\|d^j\| \leq \varepsilon$ and $-(1/t^j)\|d^j\|^2 - v^j \leq \varepsilon$, then stop: x_k is ε-optimal. Otherwise put $y^j := x_k + d^j$ and compute $g^j \in \partial f(y^j)$.

(2) (a) If **SS**(i) and **SS**(ii) hold, then make a **Serious Step**: Put $x_{k+1} := y_{k+1} := y^j$, $g_{k+1} := g^j$ and stop.

(b) If **SS**(i) holds but not **SS**(ii), then put $l^{j+1} := t^j$, $u^{j+1} := u^j$, $t^{j+1} := \frac{1}{2}(u^{j+1} + l^{j+1})$, $j := j + 1$ and go back to (1).

(c) If **NS**(i) and **NS**(ii) hold, then make a **Null Step**: Put $x_{k+1} := x_k$, $y_{k+1} := y^j$, $g_{k+1} := g^j$ and stop.

(d) If **NS**(i) holds but not **NS**(ii), then put $u^{j+1} := t^j$, $l^{j+1} := l^j$, $t^{j+1} := \frac{1}{2}(u^{j+1} + l^{j+1})$, $j := j + 1$ and go back to (1).

Let v_k, d_k, and t_k be the values, with which we leave (2.17) in case of a Serious Step or a Null Step. Then $d_k = -t_k \sum_{i \in J_k} \lambda_{k,i} g_i$ for suitable $\lambda_k = (\lambda_{k,i}) \in \Lambda(|J_k|)$ (see (2.8)). With this λ_k, we define for later use

$$z_k := \sum_{i \in J_k} \lambda_{k,i} g_i \quad \text{and} \quad \sigma_k := \sum_{i \in J_k} \lambda_{k,i} \alpha_{k,i} \,. \tag{2.18}$$

We now present the criteria that determine whether a Serious Step or a Null Step is taken (for $k = 1$ put in **NS**(ii) $z_0 := g_1$, $\sigma_0 := 0$):

SS: (i) $f(y^j) - f(x_k) < m_1 v^j$,
SS: (ii) $(g^j)^T d^j \geq m_2 v^j$ or $t^j \geq T - \nu$,
NS: (i) $f(y^j) - f(x_k) \geq m_1 v^j$,
NS: (ii) $\alpha(x_k, y^j) \leq m_3 \sigma_{k-1}$ or $|f(x_k) - f(y^j)| \leq \|z_{k-1}\| + \sigma_{k-1}$.

Discussion of SS and NS. *Ad* (2)(a) *and* (b): Condition **SS**(i) ensures, for a Serious Step, a decrease of at least m_1 times $v_k [= f_{\mathrm{CP}}(x_k; d_k)$ = decrease in the CP-model]. The first part of **SS**(ii) takes care of a substantial change in the CP-model; this follows from (we use $x_{k+1} = y_{k+1}$, $v_k < 0$ and $m_2 < 1$)

$$g_{k+1}^T d_k - \alpha_{k+1,k+1} = g_{k+1}^T d_k \geq m_2 v_k > v_k \geq g_i^T d_k - \alpha_{k,i} \quad \text{for } i \in J_k \,, \tag{2.19}$$

which implies that, after a Serious Step, the updated model (2.6) will provide some (v, d) in step $k+1 \to k+2$, which differs from the present (v_k, d_k). If the first part of **SS**(ii) does not hold (and such a change in the model cannot be guaranteed) and if t is still smaller than some upper bound T (this is taken care of by the second condition under **SS**(ii)), then we prefer to try some larger t, even if **SS**(i) holds. This motivates steps (2)(a) and (2)(b).

Ad (2)(c) *and* (d): Now suppose **NS**(i) holds. Then either f_{CP} is not yet an adequate model and/or we were too optimistic with respect to t. The obvious way out: Try some smaller t in (1); this is step (2)(d). If, however, the first condition under **NS**(ii) also holds, then a Null Step makes sense as well and we prefer this option. The reason: after such a Null Step, we get from (2.11) for $k+1$ and $i = k+1$ (use $x_{k+1} = x_k$)

$$g_{k+1}^T (x - x_k) \leq f(x) - f(x_k) + \alpha_{k+1,k+1} \,,$$

where $\alpha_{k+1,k+1} = \alpha(x_k, y_{k+1}) \leq m_3 \sigma_{k-1}$ and $m_3 < 1$. We conclude that g_{k+1} is "close" to $\partial f(x_k)$ and thus it makes sense to add g_{k+1} to the bundle at x_k. Condition **NS**(i) guarantees that this g_{k+1} contributes nonredundant information. This follows from the next inequality, which serves the same purpose as (2.19) in case of a Serious Step:

$$\begin{aligned} g_{k+1}^T d_k - \alpha_{k+1,k+1} &= f(y_{k+1}) - f(x_k) \geq m_1 v_k \\ &> v_k \geq g_i^T d_k - \alpha_{k,i} \quad \text{for } i \in J_k; \end{aligned} \tag{2.20}$$

consequently, in iteration $k+1 \to k+2$ the enriched model f_{CP} will yield some direction d which differs from the unsuccessful present d_k. This, taken together, explains one-half of (2)(c); for technical reasons (which will become clear in Proposition 2.7 below) we also make a Null Step, if **NS**(i) holds together with the second condition under **NS**(ii).

We summarize (2.17) in a flow chart (see Fig. 2.1). We state a by-product of the

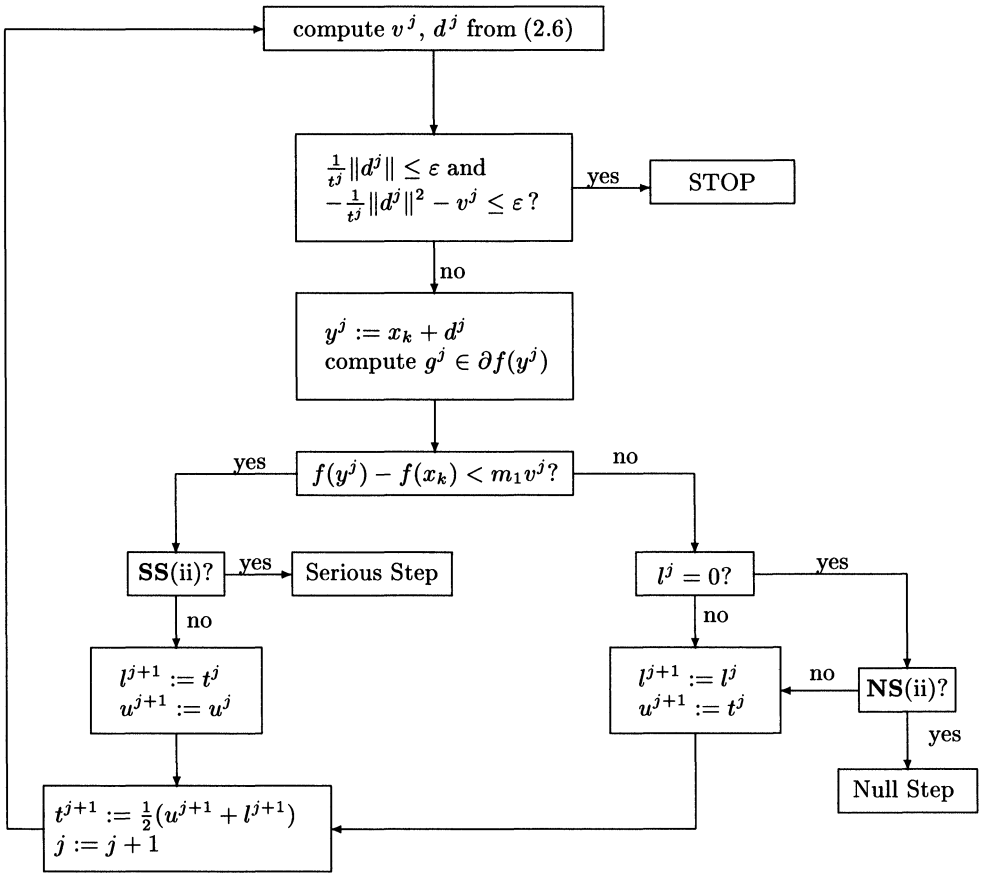

FIG. 2.1. *Flow chart for inner iteration.*

proof of Theorem 2.3 below:

(2.21) If $f(y^j) - f(x_k) < m_1 v^j$ for some j,
then one leaves (2.17) with a Serious Step.

Hence it suffices to check **NS**(ii) in Fig. 2.1 only as long as $l^j = 0$.

In our implementation of (2.17) we replace the simple bisection rule for t by a more sophisticated heuristic strategy. We choose a safeguarded variation of t, which corresponds to the change of the function value. In step (0) we choose the initial $t^1 = t_{k-1}$ only in case of a Null Step; in case of a Serious Step we choose $t^1 \ge t_{k-1}$, as in [21].

The t-variation in (2.17) corresponds to the linesearch in M1FC1. The crucial difference is: In M1FC1 one makes an a priori decision on t_k (respectively, on some dual quantity ε_k). This results in a *fixed direction* d_k and, in the line search, one "minimizes" $f(x_k + \cdot\, d_k)$. In (2.17) t is variable and we thus try *different directions* $d(t)$ when "minimizing" $f(x_k + d(\cdot))$. The examples from §4 show that this can be a decisive advantage.

Remark. Let us mention that the t-adjustment in (2.6) is actually of an *implicit* nature and one should better talk of an *implicit trust region approach* in our context. A similar implicit trust region idea was considered in a recent paper by Bell [2].

The next result supplies the actual justification for what we are doing.

THEOREM 2.3. *Iteration* (2.17) *ends after finitely many cycles, either with a Serious Step or a Null Step or the information that* x_k *is* ε*-optimal.*

Proof (by contradiction). Suppose the algorithm is an endless cycle. Then three cases can occur: (i) $l^j = 0$ for all j; (ii) $u^j = T$ for all j; (iii) neither (i) nor (ii) holds.

Ad (i): We are always on the right branch in Fig. 2.1 and thus $t^{j+1} = \frac{1}{2}(0+t^j) \downarrow 0$ and $y^j \to x_k$ as $j \to \infty$. Hence **NS**(ii) will hold for large enough j; since **NS**(i) is satisfied by construction on the right branch, (2.17) stops with a Null Step in contradiction to our assumption.

Ad (ii): Now we are always on the left branch and thus $t^{j+1} = \frac{1}{2}(t^j + T) \uparrow T$ for $j \to \infty$, i.e., **SS**(ii) holds for large enough j. Since **SS**(i) is automatically satisfied on the left branch, we will stop with a Serious Step in contradiction to our assumption.

Ad (iii): In this case $0 < l^j < u^j < T$ for all sufficiently large j and a monotonicity argument implies $l^j \uparrow t^*$ and $u^j \downarrow t^*$ for some $t^* \in (0, T)$. A continuity argument (recall (2.15)) together with **SS**(i) and **NS**(i) yields for $d^* := d(t^*)$ and $v^* := v(t^*)$

$$f(x_k + d^*) - f(x_k) = m_1 v^* . \tag{2.22}$$

Let $j(1), j(2), \cdots$ be the subsequence of indices, for which **SS**(i) holds; this is an infinite sequence since otherwise $l^{j(m)} = t^*$ for some m and then (2.22) would contradict **SS**(i). Since $l^j \uparrow t^*$, the $g^{j(i)}$ have a cluster point g^* that belongs to $\partial f(x_k + d^*)$ (we use (2.2)). Hence

$$(g^*)^T(x_k - (x_k + d^*)) \leq f(x_k) - f(x_k + d^*),$$

and, because of (2.22),

$$(g^*)^T d^* \geq m_1 v^*.$$

Now $v^* < 0$ (otherwise (2.17) would have stopped because of ε-optimality) together with $0 < m_1 < m_2$ shows $g^* d^* > m_2 v^*$. A continuity argument implies, for sufficiently large i,

$$(g^{j(i)})^T d^{j(i)} \geq m_2 v^{j(i)},$$

hence we will stop with a Serious Step in contradiction to our assumption. □

2.3. The overall algorithm. We briefly summarize the overall algorithm with *reset strategy*.

(2.23) **BT-algorithm:** Choose a starting point $x_1 \in \mathbb{R}^n$ and parameters $T > 0$, $0 < m_1 < m_2 < 1$, $0 < m_3 < 1$, $\nu > 0$, $\varepsilon \geq 0$ and an upper bound $J_{\max} \geq 3$ for $|J_k|$.

(0) Compute $f(x_1)$, $g_1 \in \partial f(x_1)$ and put $y_1 := x_1$, $J_1 := \{1\}$ and $k := 1$.

(1) INNER ITERATION: Compute x_{k+1} and g_{k+1} as in (2.17) or realize that x_k is ε-optimal (in which case we stop).

(2) If $|J_k| = J_{\max}$, then go to (3); otherwise put $J := J_k$ and go to (4).

(3) RESET: Choose $J \subset J_k$ with $|J| \leq J_{\max} - 2$ and $\max\{i \mid i \in J_k,\ \alpha_{k,i} = 0\} \in J$. Introduce some additional index $\tilde{k}$ and define with z_k, σ_k from (2.18)

$$g_{\tilde{k}} := z_k, \qquad \alpha_{k,\tilde{k}} := \sigma_k, \quad J := J \cup \{\tilde{k}\}.$$

(4) UPDATE: If the outcome of (2.17) was a Serious Step, then put

$$\alpha_{k+1,i} := \alpha_{k,i} + f(x_{k+1}) - f(x_k) - g_i^T d_k \quad \text{for } i \in J,\ \alpha_{k+1,k+1} := 0\,.$$

If the outcome of (2.17) was a Null Step, then put

$$\alpha_{k+1,i} := \alpha_{k,i} \quad \text{for } i \in J,\ \alpha_{k+1,k+1} := \alpha(x_k, y_{k+1})\,.$$

Put $J_{k+1} := J \cup \{k+1\}$ and go to (1).

Remark. We add a comment on the index $\tilde{k}$ in step (3) and the update formula in step (4).

(a) The $g_{\tilde{k}}$ defined in the reset step corresponds to the *aggregate subgradient* introduced in [18]. Usually $g_{\tilde{k}}$ will not be a subgradient at some point $y_{\tilde{k}}$ and thus $\alpha_{k,\tilde{k}}$ does not fit into the concept (2.4). It follows, however, from (2.12) that the synthetic $\alpha_{k,\tilde{k}}$ again satisfies

$$g_{\tilde{k}}^T (x - x_k) \leq f(x) - f(x_k) + \alpha_{k,\tilde{k}} \quad \text{for all } x\,,$$

which is actually what is needed from subgradients.

(b) One easily checks that for the indices i which correspond to points y_i, the update formula in (4) is in accordance with (2.4). The update strategy dispenses the need to carry along the x_i's and y_i's.

2.4. Convergence analysis. The proof technique below is largely based on ideas that go back to Kiwiel [18]. Throughout, we work with the stopping parameter $\varepsilon = 0$. Let x_k, $k = 1, 2, \cdots$, be the iterates generated by (2.23) and recall the abbreviations introduced in (2.18):

$$z_k = \sum_{i \in J_k} \lambda_{k,i} g_i \quad \text{and} \quad \sigma_k = \sum_{i \in J_k} \lambda_{k,i} \alpha_{k,i}\,.$$

In terms of z_k and σ_k, the crucial relations (2.8) and (2.9) become

$$d_k = -t_k z_k \quad \text{and} \quad v_k = -t_k \|z_k\|^2 - \sigma_k\,. \tag{2.24}$$

Further, let us denote the minimal value in (2.16) by w_k, i.e.,

$$w_k = \frac{1}{2}\|z_k\|^2 + \frac{1}{t_k}\sigma_k\,, \tag{2.25}$$

and put

$$X^* := \{x^* \in \mathbb{R}^n \mid f(x^*) \leq f(x) \text{ for all } x \in \mathbb{R}^n\}\,.$$

Finally we mention a technical assumption that we will need for our auxiliary results:

(2.26) There exists $\bar{x}$ such that $f(\bar{x}) \leq f(x_k)$ for all k.

As a foretaste of what we will prove, we summarize the result:

$f(x_k)$ converges to $\inf_x f(x)$ $(\geq -\infty)$ and,
if $X^* \neq \emptyset$, then x_k converges to some $x^* \in X^*$.

We start with the following observation.

LEMMA 2.4. *If* (2.26) *holds, then for each* $\delta > 0$ *there exists* $n_0(\delta) \in \mathbb{N}$ *such that*

$$\|\bar{x} - x_{k+1}\|^2 \leq \|\bar{x} - x_m\|^2 + \delta \quad \textit{for} \quad k \geq m \geq n_0(\delta)\,. \tag{2.27}$$

Proof. Equation (2.12) becomes, in terms of z_k and σ_k,

$$z_k^T(\bar{x} - x_k) \leq f(\bar{x}) - f(x_k) + \sigma_k$$

and, since $f(\bar{x}) \leq f(x_k)$,

$$z_k^T(\bar{x} - x_k) \leq \sigma_k\,.$$

If we put

$$\delta_k := \begin{cases} 1\,, & \text{if } k \to k+1 \text{ is a Serious Step}\,, \\ 0\,, & \text{if } k \to k+1 \text{ is a Null Step}\,, \end{cases}$$

then $x_{k+1} - x_k = \delta_k d_k = -\delta_k t_k z_k$ for all k and thus

$$-(\bar{x} - x_k)^T(x_{k+1} - x_k) = \delta_k t_k(\bar{x} - x_k)^T z_k \leq \delta_k t_k \sigma_k\,.$$

It follows that

$$\begin{aligned}\|\bar{x} - x_{k+1}\|^2 &= \|\bar{x} - x_k\|^2 + \|x_k - x_{k+1}\|^2 - 2(\bar{x} - x_k)^T(x_{k+1} - x_k) \\ &\leq \|\bar{x} - x_k\|^2 + \|x_k - x_{k+1}\|^2 + 2\delta_k t_k \sigma_k\,.\end{aligned}$$

Hence for all $m \in \mathbb{N}$ and $k \geq m$

$$\|\bar{x} - x_{k+1}\|^2 \leq \|\bar{x} - x_m\|^2 + \sum_{i=m}^{k}(\|x_i - x_{i+1}\|^2 + 2\delta_i t_i \sigma_i)\,. \tag{2.28}$$

Now consider the sum in (2.28). From $f(x_{i+1}) - f(x_i) \leq \delta_i m_1 v_i$ we obtain, for arbitrary $l > 1$,

$$f(x_1) - f(x_l) = f(x_1) - f(x_2) + f(x_2) - \cdots + f(x_{l-1}) - f(x_l) \geq -m_1 \sum_{i=1}^{l} \delta_i v_i\,,$$

and thus for $l \to \infty$ (we use (2.9), (2.8), and (2.24)),

$$\infty > f(x_1) - f(\bar{x}) \geq -m_1 \sum_{i=1}^{\infty} \delta_i v_i = m_1 \sum_{i=1}^{\infty} \delta_i (t_i \|z_i\|^2 + \sigma_i) .$$

Since $\delta_i t_i^2 \|z_i\|^2 = \|x_{i+1} - x_i\|^2$, we can continue

$$\infty > m_1 \sum_{i=1}^{\infty} \left(\frac{1}{t_i} \|x_{i+1} - x_i\|^2 + \delta_i \sigma_i \right) ,$$

and thus (we use that, by construction, $t_i \leq T$)

$$\infty > m_1 \sum_{i=1}^{\infty} (\|x_{i+1} - x_i\|^2 + \delta_i t_i \sigma_i) .$$

Consequently we can make the sum in (2.28) as small as we like by letting $m \to \infty$. This proves the assertion. □

The next lemma is an almost immediate consequence of (2.27).

LEMMA 2.5. *If* (2.26) *holds, then the* x_k *converge to some* $\tilde{x}$*, for which*

$$f(\tilde{x}) \leq f(x_k) \quad \textit{for all } k .$$

Proof. By (2.27) the x_k-sequence is bounded and has a cluster point, say $\tilde{x}$. Since, by construction, $f(x_k)$ is monotonically decreasing, we see

$$f(\tilde{x}) \leq f(x_k) \quad \text{for all } \ k .$$

Hence Lemma 2.4 applies once more (now with $\bar{x}$ replaced by $\tilde{x}$) and for given $\varepsilon > 0$ we can choose $n_0(\varepsilon/2)$ such that

$$\|\tilde{x} - x_{k+1}\|^2 \leq \|\tilde{x} - x_m\|^2 + \frac{\varepsilon}{2} \quad \text{for } k \geq m \geq n_0 \left(\frac{\varepsilon}{2} \right) .$$

Since $\tilde{x}$ is a cluster point of the x_k-sequence, there exists $\tilde{m} \geq n_0(\varepsilon/2)$ with $\|\tilde{x} - x_{\tilde{m}}\|^2 \leq \varepsilon/2$ and we end up with

$$\|\tilde{x} - x_{k+1}\|^2 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \quad \text{for all } k \geq \tilde{m} . \qquad \square$$

In the following we will show that $\tilde{x}$ from Lemma 2.5 is indeed optimal. For this aim, we prove for suitable subsequences

$$z_{k(i)} \to 0 \quad \text{and} \quad \sigma_{k(i)} \to 0 \quad \text{for } i \to \infty . \tag{2.29}$$

The optimality of $\tilde{x}$ follows from (2.12) (cf. Theorem 2.10).

We start with the crucial observation that besides x_k, the auxiliary sequences w_k, etc., are also bounded in situation (2.26).

LEMMA 2.6. *If* (2.26) *holds, then the sequences of* w_k, z_k, σ_k, d_k, y_k, g_k, *and* $\alpha_{k,k}$ $(k = 1, 2, \cdots)$ *are bounded.*

Proof. We combine (2.24) and (2.25) to see

$$0 \leq w_k = -\frac{1}{t_k} \left(v_k + \frac{1}{2t_k} \|d_k\|^2 \right) ,$$

i.e.,

$$0 \geq -w_k = \frac{1}{t_k}\left(\max_{i\in J_k}\{g_i^T d_k - \alpha_{k,i}\} + \frac{1}{2t_k}\|d_k\|^2\right).$$

Now choose $i(k) \in J_k$ such that $\alpha_{k,i(k)} = 0$ (such $i(k)$ exists because of our reset strategy); then we can continue the last inequality

$$0 \geq -w_k \geq \frac{1}{t_k}\min_{d\in\mathbb{R}^n}\left\{g_{i(k)}^T d + \frac{1}{2t_k}\|d\|^2\right\}.$$

With the minimizer $d := -t_k g_{i(k)}$ we obtain

$$0 \geq -w_k \geq -\frac{1}{2}\|g_{i(k)}\|^2. \tag{2.30}$$

The choice $\alpha_{k,i(k)} = 0$ guarantees $g_{i(k)} \in \partial f(x_k)$ (a consequence of (2.11) and (2.1)). This, together with the convergence of the x_k (Lemma 2.5) and the boundedness of the map $x \to \partial f(x)$ (see (2.2)) yields the boundedness of $\{g_{i(k)}\}_{k\in\mathbb{N}}$ and because of (2.30), the boundedness of $\{w_k\}_{k\in\mathbb{N}}$. A look at (2.25) and (2.24) convinces the reader that the sequences of z_k, σ_k,d_k, and $y_k = x_k + d_k$ are also bounded (we use $t_k \leq T$). Consequently the $g_k \in \partial f(y_k)$ are also bounded, since $\partial f(\cdot)$ is a bounded map. This, together with the convergence of the x_k and the continuity of f, finally proves the boundedness of the $\alpha_{k,k}$. □

In our two main auxiliary Propositions 2.8, 2.9 below, we will prove $v_k(i) \to 0$ (respectively, $w_k(i) \to 0$) for a suitable subsequence. A glance at (2.24) and (2.25) shows that this implies the crucial relation (2.29), provided $t_k \geq \underline{t} > 0$. The situation $t_k \to 0$ has to be treated as a special case in the next theorem. Here the role of the second condition in **NS**(ii) becomes clear: It is needed to ensure (2.29) even if $t_k \to 0$.

PROPOSITION 2.7. *Suppose* (2.26) *holds and* 0 *is a cluster point of* $\{t_k\}_{k\in\mathbb{N}}$. *Then, for suitable subsequences,*

$$\lim_{i\to\infty} z_{k(i)} = 0, \qquad \lim_{i\to\infty} \sigma_{k(i)} = 0.$$

Proof (by contradiction). Suppose there is $\delta > 0$ with

$$\|z_k\| + \sigma_k \geq \delta \quad \text{for all } k.$$

Now denote by $d_k(t)$ the solution in step (1) of (2.17) for variable t and variable k. The Lipschitz continuity of the convex f, the convergence of the x_k (Lemma 2.5) and (2.8), together with the boundedness of the g_i (Lemma 2.6) imply the existence of $L > 0$, $C > 0$, and $0 < \tilde{T} \leq T$ such that

$$|f(x_k + d_k(t)) - f(x_k)| \leq L\|d_k(t)\| \leq t\,L\,C \quad \text{for } k \quad \text{and} \quad t \leq \tilde{T}.$$

By making $\tilde{T}$ smaller, if necessary, we can guarantee that

$$|f(x_k + d_k(t)) - f(x_k)| \leq \delta \leq \|z_{k-1}\| + \|\sigma_{k-1}\| \quad \text{for } k \geq 2 \quad \text{and} \quad t \leq \tilde{T}.$$

Hence, whenever we are on the right branch in Fig. 2.1, then we will leave (2.17) with a Null Step as soon as, for the first time, $t^j \leq \tilde{T}$. Since t is increased on the left branch, we conclude from the bisection update rule for t that $t_k \geq \frac{1}{2}\tilde{T}$ for all k. This contradicts the assumption. □

It is convenient to discuss separately the case of finitely many Serious Steps and of infinitely many Serious Steps.

PROPOSITION 2.8. *Let* (2.26) *hold and suppose that one makes infinitely many Serious Steps in* (2.23). *Then for suitable subsequences,*

$$\lim_{i\to\infty} z_{k(i)} = 0, \qquad \lim_{i\to\infty} \sigma_{k(i)} = 0\,.$$

Proof. We may assume $t_k \geq \underline{t} > 0$ for all k, since otherwise the assertion follows from Proposition 2.7. Now let $\{x_{k(i)}\}_{i\in\mathbb{N}}$ be a subsequence resulting in Serious Steps, i.e.,

$$f(x_{k(i)+1}) - f(x_{k(i)}) < m_1 v_{k(i)}\,;$$

hence for $l \geq 1$ (note, $x_{k+1} = x_k$ for Null Steps)

$$f(x_{k(l)+1}) - f(x_{k(1)}) < m_1 \sum_{i=1}^{l} v_{k(i)}\,.$$

We conclude

$$f(\bar{x}) - f(x_{k(1)}) < m_1 \sum_{i=1}^{\infty} v_{k(i)}$$

and thus $0 \geq \sum_{i=1}^{\infty} v_{k(i)} > -\infty$. The assertion follows from (2.24) since $t_k \geq \underline{t} > 0$ for all k. □

PROPOSITION 2.9. *Suppose* (2.26) *holds and one makes only finitely many Serious Steps. Then for suitable subsequences*

$$\lim_{i\to\infty} z_{k(i)} = 0, \qquad \lim_{i\to\infty} \sigma_{k(i)} = 0\,.$$

Proof. Because of Proposition 2.7, we can again assume that

$$t_k \geq \underline{t} > 0 \quad \text{for all} \quad k\,. \tag{2.31}$$

Further, there exists by assumption some $\bar{k}$ with

$$x_k = x_{\bar{k}} \quad \text{for } k \geq \bar{k}\,.$$

In step (i) we will discuss the change in the minimal value w_k of (2.16) from $w_k \to w_{k+1}$ for $k \geq \bar{k}$; this is used in (ii) to show $w_k \to 0$, which proves the assertion (see (2.25)).

Ad (i): We fix some $k \geq \bar{k}$ and consider the function for $\nu \in [0,1]$

$$Q(\nu) := \frac{1}{2}\|(1-\nu)z_k + \nu g_{k+1}\|^2 + (1-\nu)\frac{1}{t_{k+1}}\sigma_k + \nu\frac{1}{t_{k+1}}\alpha_{k+1,k+1}\,.$$

A glance at (2.16) tells us that

$$w_{k+1} \leq \min\{Q(\nu) \mid 0 \leq \nu \leq 1\} =: \tilde{w}\,. \tag{2.32}$$

To unburden the notation we put

$$\Delta_k := \frac{1}{t_{k+1}} - \frac{1}{t_k}$$

and skip the subscript k in the rest of part (i) and write $+$ for $k+1$. Simple arithmetic shows

$$\begin{aligned} Q(\nu) &= \frac{1}{2}\nu^2\|z-g_+\|^2 + \nu(z^T g_+ - \|z\|^2) + \frac{1}{2}\|z\|^2 + \frac{1}{t_+}\sigma + \frac{1}{t_+}\nu(\alpha_{+,+} - \sigma) \\ &= \frac{1}{2}\nu^2\|z-g_+\|^2 + \nu(z^T g_+ - \|z\|^2) + w + \Delta\sigma + \frac{1}{t_+}\nu(\alpha_{+,+} - \sigma)\,. \end{aligned} \tag{2.33}$$

Since we only make Null Steps for $k \geq \bar{k}$, one has, as a consequence of **NS**(i),

$$g_+^T(-tz) - \alpha_{+,+} \geq m_1 v > m_2 v = m_2(-t\|z\|^2 - \sigma_k)$$

and thus

$$g_+^T z \leq -\frac{1}{t}\alpha_{+,+} + m_2\left(\|z\|^2 + \frac{1}{t}\sigma\right).$$

This inequality allows us to continue (2.33) for $\nu \in [0,1]$

$$\begin{aligned} Q(\nu) &\leq \frac{1}{2}\nu^2\|z-g_+\| + \nu\left(-\frac{1}{t}\alpha_{+,+} + m_2\|z\|^2 + m_2\frac{1}{t}\sigma - \|z\|^2\right) + w \\ &\quad + \Delta\sigma + \frac{1}{t_+}\nu(\alpha_{+,+} - \sigma) \\ &= \frac{1}{2}\nu^2\|z-g_+\|^2 - \nu(1-m_2)\left(\frac{1}{t}\sigma + \|z\|^2\right) + w \\ &\quad - \frac{1}{t}\nu(\alpha_{+,+} - \sigma) + \Delta\sigma + \frac{1}{t_+}\nu(\alpha_{+,+} - \sigma) \\ &\leq \frac{1}{2}\nu^2\|z-g_+\|^2 - \nu(1-m_2)w + w + \nu\Delta(\alpha_{+,+} - \sigma) + \Delta\sigma \\ &=: q(\nu)\,. \end{aligned} \tag{2.34}$$

With

$$(C_k =)\, C := \max\left\{\|z\|, \|g_+\|, \frac{1}{t}\sigma, 1\right\} \tag{2.35}$$

we can go on:

$$\begin{aligned} Q(\nu) &\leq q(\nu) \\ &\leq 2\nu^2 C^2 - \nu(1-m_2)w + w + \nu\Delta(\alpha_{+,+} - \sigma) + \Delta\sigma \\ &=: \bar{q}(\nu)\,. \end{aligned}$$

For the special $\bar{\nu} := (1-m_2)w/4C^2$, we obtain from (2.33) and the last inequality (note that $\bar{\nu} \in [0,1]$ since $\bar{\nu} \leq (1-m_2)(\frac{1}{2}C^2 + C)/4C^2 < 1$):

$$w_+ \leq \tilde{w} \leq \bar{q}(\bar{\nu}) = w - (1-m_2)^2\frac{w^2}{8C^2} + (1-m_2)\frac{w}{4C^2}\Delta(\alpha_{+,+} - \sigma) + \Delta\sigma\,. \tag{2.36}$$

Ad (ii): We add again the index $k(\geq \bar{k})$ to w, σ, α, t, Δ, and C from (2.35). Since we only make Null Steps for $k \geq \bar{k}$, the t_k are monotonically decreasing from $\bar{k}$ on (see Fig. 2.1) and we conclude from (2.31) that

$$\Delta_k \to 0 \quad \text{as } k \to \infty\,. \tag{2.37}$$

By Proposition 2.7 the terms z_k, g_{k+1}, σ_k in (2.35) are bounded; this, together with (2.31), implies the existence of $\bar{C}$ with $\bar{C} \geq C_k$ for all k. Inequality (2.36) simplifies to

$$w_{k+1} \leq w_k - (1-m_2)^2 \tfrac{w_k^2}{8} \bar{C}^{-2} + (1-m_2)\tfrac{w_k}{4} C_k^{-2} \Delta_k (\alpha_{k+1,k+1} - \sigma_k) + \Delta_k \sigma_k \quad \text{for } k \geq \bar{k}\,. \tag{2.38}$$

We use Lemma 2.6 once more to see that $\{w_k\}_{k\in\mathbb{N}}$ is bounded. Let a be the greatest cluster point and assume

$$w_{k(i)+1} \to a \quad \text{for } i \to \infty\,.$$

Now let b be any other cluster point of the sequence $w_{k(i)}$, i.e., for a further subsequence we have

$$w_{k(i(j))} \to b \quad \text{for } j \to \infty\,.$$

From (2.37) and (2.38) we obtain for $j \to \infty$

$$a \leq b - [(1-m_2)^2 \tfrac{1}{8} \bar{C}^{-2}] b^2 + 0\,.$$

Since, by choice, $b \leq a$, this can hold only if $a = b = 0$. This proves $w_k \to 0$ and the assertion follows from (2.25) and the boundedness of the t_k. □

Our convergence results now follow easily.

THEOREM 2.10. *If $X^* \neq \emptyset$, then x_k converges to some $x^* \in X^*$ as $k \to \infty$.*

Proof. Obviously (2.26) holds and the x_k converge to some $\tilde{x}$ (Lemma 2.5). From (2.12) we get, for each k and with z_k, σ_k from (2.18)

$$z_k^T (x - x_k) \leq f(x) - f(x_k) + \sigma_k \quad \text{for all } x\,.$$

If we fix x and choose a subsequence as in Propositions 2.7–2.9, then we obtain for $k \to \infty$

$$0 \leq f(x) - f(\tilde{x})\,.$$

Hence $x^* := \tilde{x} \in X^*$. □

The above result can be supplemented as follows.

THEOREM 2.11. *If $X^* = \emptyset$, then $f(x_k)$ converges to* $\inf\{f(x) \mid x \in X\} \in [-\infty, \infty)$.

Proof. By construction, the $f(x_k)$ are monotonically decreasing. Now suppose the assertion not to be true, i.e., for some $\bar{x}$ one has $f(\bar{x}) \leq f(x_k)$ for all k. Just as above, we conclude that $x_k \to \tilde{x} \in X^*$, which contradicts $X^* = \emptyset$. □

2.5. Piecewise linear case. For *piecewise linear* convex functions

$$f(x) := \max\{a_j^T x - b_j \mid 1 \leq j \leq m\} \quad \text{with } a_j \in \mathbb{R}^n, \qquad b_j \in \mathbb{R}\ (1 \leq j \leq m).$$

Theorem 2.10 can be refined substantially. Suppose f is bounded below (which implies $X^* \neq \emptyset$ for piecewise linear f), choose $J_{\max} := n + 2$ in (2.23), and organize the algorithm such that each g_i, $i \in J_k$, is some a_j, $1 \leq j \leq m$. For this aim we put in the reset step (3) of (2.23) $J := \{i \mid \lambda_{k,i} > 0\}$, where $d_k = -t_k \sum_{i\in J_k} \lambda_{k,i} g_i$ solves (2.6). By Carathéodory's theorem one can always find λ_k such that $|J| \leq n+1$. A quadratic programming method, which solves (2.6) with this purpose in mind, is

given in Kiwiel [19]. Further, some *Haar condition* has to be satisfied. Denote by $I(x^*)$ the set of active indices for given $x^* \in X^*$ (i.e., $I(x^*) = \{i \mid f(x^*) = a_i^T x^* - b_i\}$) and assume:

(2.39) If $I \subset I(x^*)$ and $|I| \leq n$ then the a_i, $i \in I$, are linearly independent.

Then we can establish *finite convergence* for our algorithm.

THEOREM 2.12. *Suppose f is piecewise linear, bounded below, and* (2.39) *holds. Then $x_k \in X^*$ for some $k \in \mathbb{N}$.*

We omit the proof, which the reader can easily copy from the above discussion and the treatment of this topic in Chapter 2 of Kiwiel [18].

The numerical results from §4 will display the finite convergence convincingly.

3. BT-algorithm: The nonconvex case. We discuss the modifications necessary for nonconvex f. Throughout this section we assume that f is locally Lipschitzian and

(3.1) weakly semismooth,

i.e., the directional derivative $f'(x;d) := \lim_{t \downarrow 0} t^{-1}[f(x+td) - f(x)]$ exists for all x and d, and $f'(x;d) = \lim_{t \downarrow 0} g(x+td)^T d$ where $g(x+td) \in \partial f(x+td)$.

3.1. Model and algorithm. For nonconvex f, the subgradient inequality (2.1) does not hold and the $\alpha_{k,i}$ may become negative. As a consequence, $f_{\mathrm{CP}}(x_k;\cdot)$ is no longer an approximation of $f(x_k+\cdot) - f(x_k)$ from below; in particular, usually $f_{\mathrm{CP}}(x_k;0) = \max\{-\alpha_{k,i}\} > f(x_k+0) - f(x_k)$. To cope with this difficulty we follow a strategy (also used in M1FC1) and replace $\alpha_{k,i}$ by

$$\beta_{k,i} := \beta(x_k, y_i) := \max\{\alpha_{k,i},\, c_0\|x_k - y_i\|^2\};$$

here c_0 is a fixed small positive real (and $c_0 := 0$ for convex f). By construction, $\beta_{k,i} \geq 0$ and the modified model

$$f_{\mathrm{CP}}(x_k;d) := \max_{i \in J_k} \{g_i^T d - \beta_{k,i}\} \tag{3.2}$$

coincides again with $f(x_k+d) - f(x_k)$, at least at $d = 0$. The $\beta_{k,i}$ copy part of the role of the $\alpha_{k,i}$ in §2: Whenever y_i is "far away" from the current iterate x_k, then $\beta_{k,i}$ is large and thus g_i only plays a minor role in (3.2). However, we have to admit that the above f_{CP} is a much less satisfactory model in the nonconvex case.

Now replace, in §§2.1–2.3, the $\alpha_{k,i}$ by the new weights $\beta_{k,i}$. This does not change the character of (2.6) and (2.16), and thus Lemma 2.1 and the duality between (2.6) and (2.16) remain true. For (2.12), however, convexity was essential and as a consequence (2.14) and the corresponding criterion in (2.17)(1) no longer imply the ε-optimality for x_k. For nonconvex f the condition

$$\left\| \sum_{i \in J_k} \lambda_i g_i \right\| \leq \varepsilon \quad \text{and} \quad \sum_{i \in J_k} \lambda_i \beta_{k,i} \leq \varepsilon$$

merely says that 0 "lies up to ε" in the convex hull of certain $g_i \in \partial f(y_i)$ for which the "y_i are not far away from x_k" (since $\sum_{i \in J_k} \lambda_i \beta_{k,i} \leq \varepsilon$). This corresponds to "almost" stationarity in smooth optimization.

Iteration (2.17) requires two modifications. First, part (iii) in the proof of Theorem 2.3 does not carry over to nonconvex f. This difficulty is easily bypassed: We simply omit condition **SS**(ii) in step (2)(a) in (2.17) and skip step (2)(b). This does not affect the convergence analysis since the purpose of **SS**(ii) in (2.17) was of merely numerical nature.

Second, for nonconvex f, we have to add in (2)(c) the further Null Step condition

NS: (iii) $g^{j^T} d^j - \beta_{k,j} \geq m_2 v^j$.

NS(iii) guarantees that, after a Null Step, the updated model provides a direction d that differs from the unsuccessful previous one. This change in the direction played a crucial role in the convergence analysis (see, e.g., the proof of Proposition 2.9). For convex f, condition **NS**(iii) is automatically satisfied whenever we are on the right branch in Fig. 2.1; cf. (2.20). This is not true for nonconvex f and we have to add this as an additional condition. Unfortunately, this supplementary **NS**(iii) leads to a serious drawback of our method. For nonconvex f we can no longer guarantee the existence of $\tilde{T} > 0$ in the proof of Proposition 2.7 such that **NS**(ii) together with **NS**(iii) holds on the right branch of Fig. 2.1 for $t^j \leq \tilde{T}$. As a consequence we cannot assure any more that $z_k \to 0$ and $\sigma_k \to 0$ for the special case $t_k \to 0$, which had to be separated from the proof of Propositions 2.8 and 2.9.

As things stand now, we can propose only the following emergency exit. We add to **NS**(i) and **NS**(ii) the additional condition **NS**(iii) and split (2)(d) of (2.17) in two branches. Suppose **NS**(i) holds but **NS**(iii) does not: If the second condition of **NS**(ii) is not satisfied, then we are allowed to choose a smaller t^{j+1}; if the second condition of **NS**(ii) holds, then we make a *linesearch* along d^j, just as in M1FC1. More precisely, this yields the following.

(3.3) **Inner iteration** $x_k \to x_{k+1}$**:**

(0) Choose $t^1 := t_{k-1}$. Set $l^1 := 0$, $u^1 := T$, and $j := 1$.

(1) Compute the solution $(v^j, d^j) = (v(t^j), d(t^j))$ of (2.6) with $\alpha_{k,i}$ replaced by $\beta_{k,i}$. If $(1/t^j)\|d^j\| \leq \varepsilon$ and $-(1/t^j)\|d^j\|^2 - v^j \leq \varepsilon$, then stop: x_k is almost stationary. Otherwise put $y^j := x_k + d^j$ and compute $g^j \in \partial f(y^j)$.

(2) (a) If **SS**(i) holds, then make a **Serious Step**: Put $x_{k+1} := y_{k+1} := y^j$, $g_{k+1} := g^j$ and stop.

(b) If **NS**(i), **NS**(ii), and **NS**(iii) hold, then make a **Null Step**: Put $x_{k+1} := x_k$, $y_{k+1} := y^j$, $g_{k+1} := g^j$ and stop.

(c) If **NS**(i), **NS**(ii) hold but **NS**(iii) does not, then:

(i) if the second part of **NS**(ii) holds, then put $d_k := d^j$, $v_k := v^j$ and make a linesearch along $x_k + sd_k$, $s \geq 0$,

(ii) otherwise put $u^{j+1} := t^j$, $l^{j+1} := l^j$, $t^{j+1} := \frac{1}{2}(u^{j+1} + l^{j+1})$, $j := j + 1$ and go back to (1).

(d) If **NS**(i) holds but **NS**(ii) does not, then put $u^{j+1} := t^j$, $l^{j+1} := l^j$, $t^{j+1} := \frac{1}{2}(u^{j+1} + l^{j+1})$, $j := j + 1$ and go back to (1).

For weakly semismooth f (see (3.1)) the linesearch ends up in finitely many steps with a stepsize $s_k \geq 0$ such that in $y_{k+1} := x_k + s_k d_k$ and $g_{k+1} \in \partial f(y_{k+1})$, either the (short) serious criterion

SSS: (i) $f(y_{k+1}) - f(x_k) < m_1 s_k v_k$,

SSS: (ii) $g_{k+1}^T d_k \geq m_2 v_k$

is satisfied or **NS**(iii) and the first part of **NS**(ii) hold but **SSS**(i) is not satisfied. In case **SSS**(i) and **SSS**(ii) hold, we put $x_{k+1} := y_{k+1}$ and add g_{k+1} to the bundle; if **NS**(iii) and the first part of **NS**(ii) hold, we make a Null Step. All details concerning the linesearch can be found in Lemaréchal [22]. Semismoothness is a property with respect to halflines and this explains the success of a linesearch. The heart of our argument was simply not to restrict the search to a halfline; we wanted to work with various directions. Hence we consider a linesearch only as an emergency step that is against the spirit of our approach. And, indeed, if our method runs into numerical troubles, then usually this is because we had to switch to a linesearch that ends in a collapse.

In the overall algorithm (2.23) the updating of the $\beta_{k,i}$, together with the reset strategy, has to be adapted to the new situation. We do this just as in Kiwiel's aggregate subgradient method, where one avoids again the storing of the previous x_i and y_i; these technicalities are skipped here.

We mention that linear constraints can be added in (2.6) and (2.16) without major difficulties. Proposals on how to handle nonlinear constraints in the bundle framework have been made, e.g., by Kiwiel [18].

3.2. Convergence analysis. Suppose we use the above definition of $\beta_{k,i}$ and do not use a reset strategy. It is easily verified that the inner iteration (3.3) is again a finite process. Lemmas 2.4 and 2.5 rely decisively on the subgradient inequality and do not carry over to nonconvex f. Hence the statement of Lemma 2.5 now becomes an assumption:

$$\{x_k\}_{k\in\mathbb{N}} \text{ is bounded.} \tag{3.4}$$

With (2.26) replaced by (3.4), Lemma 2.6 remains true. The same holds for Propositions 2.7–2.9; of course, eventual linesearch steps have to be taken into account. Only the proof of Proposition 2.9 requires some technical modifications. For nonconvex f and $\varepsilon = 0$, we get the following convergence result (for details, see Schramm [37]).

THEOREM 3.1. *If f is weakly semismooth, bounded below, and* (3.4) *holds, then there exists a cluster point $\bar{x}$ of the sequence $\{x_k\}_{k\in\mathbb{N}}$ such that $0 \in \partial f(\bar{x})$.*

4. Numerical examples and applications. The above concept was implemented in FORTRAN 77 as **BTC** for convex f and **BTNC** for nonconvex f. The implemented reset strategy and a "safeguarded weighting technique" go back to proposals by Kiwiel [18], [21]. Further, we use a subroutine due to Kiwiel (see [19]) to solve the dual quadratic programming problem at each iteration.

Section 4.1 reports our experience with a collection of (non)convex academic test problems. In §4.2 we compute dual bounds for traveling salesman problems, and in §4.3 we deal with minimax eigenvalue problems for matrices coming from special graphs. In §4.4, finally, we present the results for a nonconvex and nonsmooth optimal design problem: The maximization of the area of contact for the deflection of a clamped beam.

All computations were done on a HP9000/330, respectively, on a VAX8600.

4.1. "Academic" testexamples. Convex examples. Table 4.2 presents a comparison of BTC with M1FC1 [27] for a collection of classical convex test examples listed in Table 4.1. The following abbreviations are used in Tables 4.1–4.4.

Dim	dimension of the problem,
f^*	(known) optimal value,
niter	number of iterations,
$\# f/g$	number of function/subgradient-evaluations,
f	computed approximation of f^*.

Many of the test examples are described in detail in [25] or [42]. Test function Mifflin1 has been communicated to us by Mifflin [32]:

$$f(x) = -x_1 + 20 \max\{x_1^2 + x_2^2 - 1, 0\},$$

with starting point $(0.8, 0.6)^T$, minimum $x^* = (1, 0)^T$, and $f(x^*) = -1$.

TABLE 4.1
List of convex examples.

Nr.	Problem	Dim	f^*
1	CB2 [3]	2	1.952225
2	CB3 [3]	2	2
3	DEM [6]	2	-3
4	QL [42]	2	7.2
5	LQ [42]	2	$-\sqrt{2}$
6	Mifflin1 [32]	2	-1
7	Mifflin2 [15]	2	-1
8	Mak [29]	3	-132.0608
9	Rosen [3]	4	-44
10	Shor [39]	5	22.60016
11	Maxquad [25]	10	-0.8414084
12	Maxq [37]	20	0
13	Maxl [37]	20	0
14	Goffin [12]	50	0
15	TR48 [25]	48	-638565

Since BTC and M1FC1 proved to be strongly superior to all subgradient variants, which we tested, we restrict our comparison to BTC and M1FC1. In BTC we put $m_1 := 0.1$, $m_2 := 0.2$, $m_3 := 0.9$, $\varepsilon \leq 10^{-4}$. Furthermore we take $k$max(maximal number of subgradients):= 5 for CB2, CB3, DEM, QL, LQ, and $k$max := 10 for Mifflin1, Mifflin2, Rosen/Suzuki, Shor, Maxquad; for higher-dimensional examples we use $k$max := 200. The parameters in M1FC1 were chosen correspondingly. We mention that BTC works in double precision, whereas M1FC1 requires function- and subgradient-evaluations only in single precision. The function value and the corresponding subgradient are computed in one subroutine.

In all examples, BTNC reached the required accuracy; the same holds for M1FC1 apart from the data marked by "*," where M1FC1 broke down with linesearch difficulties. Obviously BTC often shows a better performance for the discussed examples than M1FC1. We note that results similar to ours are reported by Kiwiel [21] for his proximity control algorithm, which is closely related to our approach.

In §2.5 we discussed "finite convergence" of BTC for convex piecewise linear f. This finite convergence can be observed for the piecewise linear examples of Table 4.2. As an example, we give the finite behaviour for Goffin's test function [12]

$$f(x) = 50 \max_{1 \leq i \leq 50} x_i - \sum_{i=1}^{50} x_i ,$$

TABLE 4.2
Convex examples.

	BTC			M1FC1		
Nr.	niter	$\#f/g$	f	niter	$\#f/g$	f
1	13	16	1.952225	11	31	1.952253
2	13	21	2.000000	12	44	2.001415
3	9	13	-3.000000	10	33	-3.000000
4	12	17	7.200009	12	30	7.200018
5	10	11	-1.414214	16	52	-1.141420
6	49	74	-1.000000	143	281	-0.999967
7	6	13	-1.000000	30	71	-0.999993
8	24	28	-132.0608	3	5	-132.0608
9	22	32	-43.99998	22	61	-43.99998
10	29	30	22.60016	21	71	22.60018
11	45	56	-0.8414083	29	69	-0.8413589
12	125	128	0.0	144	207	0.0
13	74	84	0.0	138	213	0.0
14	51	53	0.0	72	94	0.00010
15	165	179	-638565.0	163	284	-633625.5*

with starting point $x^i = i - 25.5$, $i = 1, \cdots, 50$, and optimal value 0. We use the notation:

niter	number of iterations,
ncomp	number of function/subgradient-evaluations,
f	function value at the current iterate,
gn	$\|\sum_{i \in J_k} \lambda_{k,i} g_i\|$,
alpha	$\sum_{i \in J_k} \lambda_{k,i} \alpha_{k,i}$.

```
BT-Algorithm  --  Goffin
========================

 niter  ncomp        f              gn             alpha
   1      1    .12250000E+04   .49497475E+02   .00000000E+00
   2      2    .12250000E+04   .34641019E+02   .24989898E+02
   3      5    .11497487E+04   .33243153E+02   .20320902E+03
   4      6    .11497487E+04   .31880038E+02   .65908627E+02
   5      7    .10744975E+04   .27719626E+02   .23738585E+03
   6      8    .10744975E+04   .29102567E+02   .99746186E+02
   7      9    .99924621E+03   .24138528E+02   .24395184E+03
   8     10    .99924621E+03   .26342143E+02   .12550253E+03
   9     11    .92399495E+03   .21277728E+02   .23862048E+03
  10     12    .92399495E+03   .23604747E+02   .14317764E+03
   .      .          .               .               .
   .      .          .               .               .
  20     22    .57248737E+03   .89001539E+01   .18329243E+03
  30     32    .41997475E+03   .65860451E+01   .22510361E+03
  40     42    .24120581E+03   .42541465E+01   .19390350E+03
   .      .          .               .               .
   .      .          .               .               .
  45     47    .18869318E+03   .30403588E+01   .17830897E+03
  46     48    .18869318E+03   .29196707E+01   .17519938E+03
```

```
 47    49    .16243687E+03    .22448944E+01    .16184848E+03
 48    50    .16243687E+03    .21904615E+01    .15727798E+03
 49    51    .16243687E+03    .19993010E+01    .15532349E+03
 50    52    .13618055E+03    .11116780E+01    .13312204E+03
 51    53    .67968547E-12    .37861010E-14    .60867311E-12
convergence
```

The last step yields a "jump" to the optimum by adding the decisive subgradient information.

Nonconvex examples. In Table 4.4 we compare BTNC to M1FC1 (which can also deal with nonconvex f) for the problems given in Table 4.3. Here "Rosb" is the differentiable Rosenbrock example.

TABLE 4.3
List of nonconvex examples.

Nr.	Problem	Dim	f^*
1	Cres [18]	2	0
2	Mad [28]	2	0.6164324
3	Mabs [30]	2	0
4	ℓ_1 [8]	3	7.894231
5	ℓ_1a [8]	6	0.559814
6	Rosb	2	0

TABLE 4.4
Nonconvex examples.

	BTNC			M1FC1		
Nr.	niter	$\#f/g$	f	niter	$\#f/g$	f
1	24	27	$0.944280 \cdot 10^{-6}$	31	93	$0.225317 \cdot 10^{-5}$
2	21	22	0.6164324	17	41	0.6164330
3	30	39	$0.444089 \cdot 10^{-14}$	37	88	$0.111921 \cdot 10^{-7}$
4	21	23	7.894231	16	39	7.894232
5	73	78	0.559814	116	318	0.559814
6	79	88	$0.130389 \cdot 10^{-11}$	70	121	$0.243610 \cdot 10^{-6}$

Again, our bundle trust region version BTNC yields better results than M1FC1; this is further confirmed by some test runs done by Schittkowski [36]. Since, however, our experience with nonconvex problems is still rather limited, we do not claim this to be a final statement.

4.2. Traveling salesman problems. In many practical applications one has to solve a problem which can be phrased as a (symmetric) *traveling salesman problem*: Given a complete graph $K_n = (V, E)$ and distances c_{ij} for each edge $ij \in E$ (with $c_{ij} = c_{ji}$), find a tour T^* with length $c(T^*)$ as small as possible. Since problems of this type often appear in tremendous size (e.g., drilling problems with several thousands of knots), it is generally not possible to solve them exactly. Widely used tools in combinatorial optimization are therefore heuristics, which compute an approximate solution T rather quickly. To judge the quality of such a tour T, it is important to know a lower bound for the length $c(T^*)$ of the optimal tour. Such a bound can be found via the 1-*tree relaxation* described in Held and Karp [17]. We can formulate

the TSP as a linear problem of the form

$$\min\{\langle c, x\rangle \mid Ax = a,\ Bx \leq b, x_i \in \{0,1\}\}.$$

The following weak duality relation holds:

$$c(T^*) = \min\{c(T) \mid T \text{ is a tour}\} \geq \max\{\phi(\lambda) \mid \lambda \in \mathbb{R}^n\}, \tag{4.1}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\phi(\lambda) := \min\{\langle c, x\rangle + \langle \lambda, Ax - a\rangle \mid Bx \leq b,\ x_i \in \{0,1\}\}.$$

Without the binary constraints $x_i \in \{0,1\}$, one has even equality in (4.1). For our TSPs the gap was never greater than 1–3 percent. Hence we can find a good lower bound by maximizing the function ϕ which, as minimum of finitely many linear functions in λ, is nonsmooth, concave, and piecewise linear. It is known from combinatorics that $\phi(\lambda)$ can be computed via the length $\tilde{c}$ of a so-called minimum spanning 1-tree $x(\lambda)$ for our graph with the new distances $\tilde{c}_{ij} := c_{ij} + \lambda_i + \lambda_j$; it holds $\phi(\lambda) = \tilde{c} - 2\sum_{i=1}^n \lambda_i$. There are efficient algorithms to compute such a tree and thus $\phi(\lambda)$; we used the Prim algorithm. Simple subgradient calculus shows that this tree $x(\lambda)$ provides us, for free, with a subgradient of ϕ at λ: $\phi(\lambda) = \langle c, x(\lambda)\rangle + \langle \lambda, Ax(\lambda) - a\rangle$ and as a byproduct we obtain a subgradient of ϕ at λ

$$g(x(\lambda)) := A\,x(\lambda) - a \in \partial\phi(\lambda).$$

The components of $g(x(\lambda))$ are just the degrees of the knots of the 1-tree as follows:

$$g(x(\lambda))_i = \text{degree}(i) - 2, \qquad i = 1, \cdots, n.$$

Thus we are precisely in the framework (1.4) and can apply BTC or M1FC1.

The following subgradient variant

$$x_{k+1} := x_k - M\rho^k(\alpha_k g_k + (1-\alpha_k)g_{k-1})/\|\alpha_k g_k + (1-\alpha_k)g_{k-1}\| \tag{4.2}$$

with fixed $M > 0$, $0 < \rho < 1$ and $0 < \alpha_k \leq 1$ for all $k \in \mathbb{N}$ is currently the standard method in the TSP context. For the choice $\alpha_k = 1$, $k \in \mathbb{N}$, one can establish convergence with geometric convergence speed (with factor $M\rho$) of the x_k to some limit which, however, need not be optimal. We believe that Table 4.6 will convince the reader to also consider more sophisticated methods like BTC or M1FC1. Presented are the results for a collection of synthetic examples (Krolak1, $\cdots$, Krolak5) and for some TSPs which come from drilling problems. Table 4.5 gives a list of the problems we treated. "Dim" in the third column is the number of knots (i.e., the dimension of our optimization problem). The fourth column gives the length of a tour, which is considered a good one (it is not known whether this tour is optimal).

Table 4.6 shows the results. Here "lb" is the lower bound which we obtained from the three methods; for (4.2) we tried several M's and ρ's and report our best results. Finally, "%" gives the remaining gap in percentage. In BTC we take $m_1 := 0.01$, $m_2 := 0.2$, and $m_3 := 0.9$. BTC was stopped when the stopping criterion was satisfied with $\varepsilon := 10^{-4}$ for the smaller problems, respectively, $\varepsilon := 10^{-2}$ for the larger ones. Also, for M1FC1 we could satisfy a corresponding stopping criterion, apart from a few runs where the code broke down with linesearch difficulties close to the optimal point. The subgradient method was stopped when we observed no further progress

TABLE 4.5
List of traveling salesman problems.

Nr.	Problem	Dim	Tour
1	KROL1	100	21282
2	KROL2	100	22141
3	KROL3	100	20749
4	KROL4	100	21294
5	KROL5	100	22068
6	TSP442	442	5069
7	TSP1173	1173	57323
8	V362	362	1966
9	V614	614	2312
10	V1167	1167	5657
11	V2116	2116	6786

TABLE 4.6
Traveling salesman problems.

	Subgradient Method			M1FC1			BTC		
Nr.	$\#f/g$	lb	%	$\#f/g$	lb	%	$\#f/g$	lb	%
1	194	20929	1.66	103	20938	1.62	58	20938	1.62
2	202	21648	2.23	606	21753	1.75	233	21833	1.39
3	264	20451	1.44	156	20473	1.33	79	20473	1.33
4	116	20951	1.61	326	21110	0.86	118	21142	0.71
5	183	21779	1.31	292	21784	1.29	136	21799	1.22
6	229	5043	0.51	248	5033	0.71	378	5051	0.36
7	78	56351	1.70	621	56193	1.97	399	56386	1.63
8	161	1941	1.27	360	1930	1.83	285	1942	1.22
9	129	2253	2.55	255	2250	2.68	179	2254	2.51
10	141	5579	1.38	442	5564	1.64	506	5580	1.36
11	109	6599	2.76	668	6579	3.05	713	6606	2.65

in the leading digits; obviously we have convergence to a nonoptimal point, e.g., for KROL2.

Finite convergence was again observed for many of the TSPs (recall that ϕ is concave and piecewise linear). Below we give the result for KROL1.

```
BT-Algorithm  --  KROL1
=======================

  niter   ncomp        f              gn             alpha
    1       1    .19094198E+05   .66332496E+01   .00000000E+00
    2       3    .19370920E+05   .52129811E+01   .63650738E+03
    3       4    .19654650E+05   .41342650E+01   .72867439E+03
    4       5    .20150557E+05   .35600250E+01   .52416612E+03
    5       6    .20295988E+05   .23095560E+01   .69738857E+03
    .       .          .               .               .
   10      11    .20551025E+05   .14857001E+01   .30072466E+03
    .       .          .               .               .
   20      21    .20823536E+05   .71434597E+00   .11684214E+03
```

```
 .     .          .                .                .
30    31    .20893782E+05    .47437971E+00    .69645027E+02
 .     .          .                .                .
40    42    .20926464E+05    .58577127E+00    .88800444E+01
 .     .          .                .                .
50    52    .20934447E+05    .17380992E+00    .32980892E+01
51    53    .20935444E+05    .13554464E+00    .23477955E+01
52    54    .20935905E+05    .99386622E-01    .19572159E+01
53    55    .20935905E+05    .96884277E-01    .19192909E+01
54    56    .20937609E+05    .15142964E-01    .31555350E+00
55    57    .20937609E+05    .51250884E-02    .31600774E+00
56    58    .20937926E+05    .39299811E-15    .63493827E-11
convergence
```

4.3. Minimizing the maximal eigenvalue. Often an application requires the solution of the subproblem

$$(\mathrm{P}_\lambda) \qquad \text{minimize } f(x) := \lambda_{\max}(A(x))\,;$$

here $A(\cdot)$ is a real symmetric $m \times m$-matrix, which depends linearly on $x \in \mathbb{R}^n$, and $\lambda_{\max}(A(x))$ denotes the maximal eigenvalue of $A(x)$. The following properties hold (see, e.g., [5]):

- f is convex;
- f is nonsmooth at x, if the maximal eigenvalue $f(x)$ has multiplicity greater than 1;
- if u is eigenvector of $A(x)$ for the eigenvalue $f(x)$ and $\|u\|_2 = 1$, then a subgradient of f at x can be easily computed from the dyadic product uu^T.

Hence we are again in the situation (1.4) and can attack (P_λ) with bundle-type methods.

We encountered such problems in connection with

(i) stable sets of graphs,

(ii) experimental design (see Gaffke and Mathar [10]).

First, numerical steps for (ii) are reported in [1]. Here we consider (i) more closely. The theoretical background is discussed in detail in a book by Grötschel, Lovász, and Schrijver [14], who brought this subject to our attention. Let $G = [V, E]$ be a graph, $w = (w_1, \cdots, w_{|V|})^T$ a vector in $\mathbb{R}^{|V|}$ with nonnegative components, and put $\bar{w} := (\sqrt{w_1}, \cdots, \sqrt{w_{|V|}})^T$. We want to compute the so-called *theta-function* $\vartheta(G; \cdot)$,

$$\vartheta(G; w) := \min_{A \in M} \lambda_{\max}(A + W)\,,$$

where $W = \bar{w}\bar{w}^T$ and

$$M := \{B \mid B \text{ symmetric } n \times n\text{-matrix}\,,\ b_{ii} = 0 \text{ for } i \in V,\ b_{ij} = 0 \text{ for } i, j \text{ nonadjacent}\}\,.$$

The theta-function is the support function of the convex set $\mathrm{TH}(G)$ (a set that contains the convex hull of the incidence vectors of all stable sets of G). Its value is known for some special cases (let $w = (1, \cdots, 1)^T$):

(a) If G is a circle with an odd number n of knots, then

$$\vartheta(G; w) = \frac{n \cos \dfrac{\pi}{n}}{1 + \cos \dfrac{\pi}{n}}\,;$$

(b) if G is an Erdös–Ko–Rado graph $K(n,r)$, then

$$\vartheta(K(n,r);w) = \binom{n-1}{r-1}.$$

To compute $\vartheta(G;w)$, we have to solve the nonsmooth convex problem

$$(\mathrm{P}^{\vartheta}) \qquad \text{minimize} \quad \lambda_{\max}(A+W) \quad \text{subject to } A \in M.$$

Since W is constant and the constraints only require A to be symmetric and some components of A to be zero, (P^{ϑ}) can be phrased as an unconstrained minimization problem of form (P_λ), where $A = A(x)$ and $x \in \mathbb{R}^m$ corresponds to the free components of A. The dimension is equal to

$$m = \frac{n(n-1)}{2} - |\{(ij) \mid i,j \text{ nonadjacent } \}|.$$

Table 4.7 shows some results for (a). Note that for circles the dimension of the optimization problem equals the number of knots n. The starting point is always $(-1,\cdots,-1)^T$ and $k\max := 20$. The value ϑ in the second column is the precise value of $\vartheta(G;w)$.

TABLE 4.7
Odd circles.

		BTC			M1FC1		
n	ϑ	niter	$\#f/g$	f	niter	$\#f/g$	f
17	8.42701	19	23	8.42705	28	70	8.42706
23	11.44619	26	30	11.44619	42	104	11.44626
39	19.46833	42	44	19.46833	37	102	19.46837
55	27.47756	49	49	27.47756	52	132	27.47756
111	55.48889	50	50	55.48889	58	134	55.48897

Table 4.8 gives the corresponding results for (b). Here we put $k\max := 20$ for dimension $M \le 50$ and $k\max := 50$ otherwise. In the last two examples (*), M1FC1 breaks down with linesearch difficulties.

The results show clearly what was observed above: For convex f the code BTC seems to be superior to M1FC1.

TABLE 4.8
Erdös–Ko–Rado graphs.

				BTC			M1FC1		
n	r	Dim	ϑ	niter	$\#f/g$	f	niter	$\#f/g$	f
5	2	15	4	28	28	4.000003	25	52	4.000047
6	2	45	5	33	33	5.000013	33	62	5.134571
10	2	630	9	63	63	9.000008	75	95	9.000071
9	3	840	28	63	65	28.000095	28	36	53.000003*
10	4	1575	84	125	128	84.000409	37	48	163.333332*

Table 4.9 discusses the improvement of BTC, if we add all subgradients, which we obtain from a system of orthonormal eigenvectors for $f(x)$, to the bundle at x. We considered eigenvalues as equal if they differ in value less than 10^{-9}. The extreme improvement is probably due to some hidden structure of the problem, since the same technique applied to random graphs only leads to no improvement.

TABLE 4.9
Circles—enlarged information.

	BTC			modified BT		
n	niter	$\#f/g$	f	niter	$\#f/g$	f
17	12	15	8.42701	3	6	8.42701
23	29	29	11.44619	2	3	11.44619
39	30	33	19.46833	3	3	19.46833
55	49	49	27.47756	3	3	27.47756
111	50	50	55.48889	3	7	55.48888

4.4. Maximization of the contact area between a clamped beam and a rigid obstacle. Let $y(s)$ for $0 \le s \le 1$ be the deflection (*state variable*) of a clamped beam under the load $u(s)$ (*design variable*). The total amount of load $\int_0^1 u(s)ds$ is given and $u(s)$ is bounded from above by some β for each s; further, the deflection of the beam is limited from below by some rigid obstacle $g(s)$. Then the aim is to find a load density such that the beam comes as "close" to the obstacle as possible. In an abstract setting the problem becomes

$$\text{(4.3)}\quad \text{minimize } \int_0^1 (y(s)-g(s))^2 ds \quad \text{subject to}$$
$$\langle Ay, z-y\rangle \ge \langle B(x), z-y\rangle \quad \text{for all } z \in K, \qquad x \in X_{ad} \qquad (x \in X,\ y \in Y).$$

Here $X := L^2((0,1))$ and $Y := H_0^2((0,1))$ are the control space and state space, respectively. $X_{ad} := \{x \in L^\infty((0,1)) \mid 0 \le x(s) \le \beta$ almost everywhere in $(0,1)$ and $\int_0^1 x(s)ds = M\}$ and $K := \{z \in Y \mid z(s) \ge g(s)$ almost everywhere in $(0,1)\}$ are the sets of feasible controls and admissible state variables. The operator $A : Y \to Y'$ is defined by $Ay = y^{iv}$ and $B : X \to H^{-2}((0,1))$ is the natural embedding. The variational inequality (4.3) assigns to a given load x the deflection y; it is known that (4.3) can be rewritten as a quadratic programming problem

$$\text{(4.4)}\qquad y \in \arg\min_{z\in K} \frac{1}{2}\langle z, Az\rangle - \langle B(x), z\rangle .$$

The above problem is thoroughly discussed in a more general framework in [16] and [33]. The discretization below follows these references.

For a numerical treatment we divide the interval $[0,1]$ into n equidistant subintervals of length $1/n$ and consider design functions which are a constant x_i on each subinterval i. For the controls we use a standard finite-element technique and work with functions

$$y(s) = \sum_{i=1}^{2n-2} y_i \varphi_i(s),$$

where the φ_i are third-order polynomials chosen such that $y(i/n) = y_{2i-1}$ and $y'(i/n) = y_{2i}$ for $i = 1, \cdots, n-1$. In this framework K becomes $\{z \in \mathbb{R}^{2n-2} \mid z_{2i-1} \ge g(i/n)$ for $i = 1, \cdots, n-1\}$ and X_{ad} reduces to $\{x \in \mathbb{R}^n \mid \sum(x_i/n) = M,\ 0 \le x_i \le \beta$ for $i = 1, \cdots, n\}$. With the positive-definite $(2n-2) \times (2n-2)$ rigidity matrix H (built up from terms $\int \varphi_i'' \varphi_j''$) and the $(2n-2)$-vector $b(x)$ (with elements $\int x\varphi_i$), the discretized

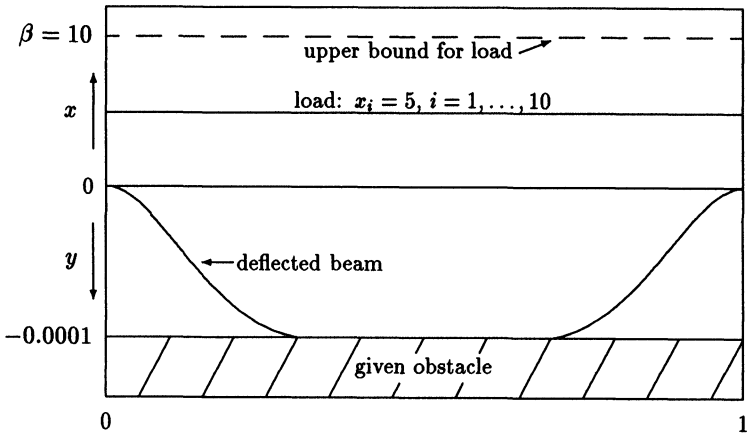

FIG. 4.1. *Beam (constant load).*

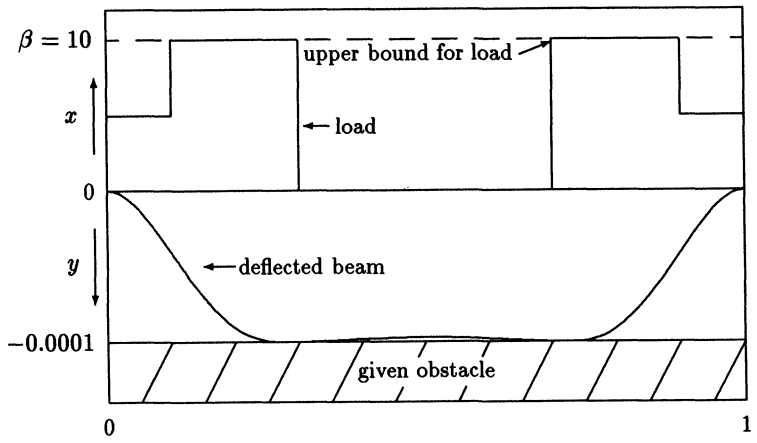

FIG. 4.2. *Beam (optimized load).*

problem becomes

$$\text{(4.5)} \quad \text{minimize} \int_0^1 \sum_{i=1}^{2n-2} (y_i \varphi_i(s) - g(s))^2 ds \quad \text{subject to}$$

$$\text{(4.6)} \quad y \in \arg\min_{z \in K} \frac{1}{2} z^T H z - b(x)^T z, \qquad x \in X_{ad}, \qquad (x \in \mathbb{R}^n,\ y \in \mathbb{R}^{2n-2}).$$

Now let $y(x) \in \mathbb{R}^{2n-2}$ denote the unique solution of (4.6) for given $x \in \mathbb{R}^n$ and write $l(y)$ for the integral in (4.5). Then our problem becomes

$$\text{(4.7)} \qquad \text{minimize } f(x) := l(y(x)) \quad \text{subject to } x \in X_{ad}.$$

Obviously f is not convex; further, because of the constraint in (4.6), the function $y(\cdot)$ (and thus also f) depends in a nonsmooth way on x. To compute $f(x)$ we have

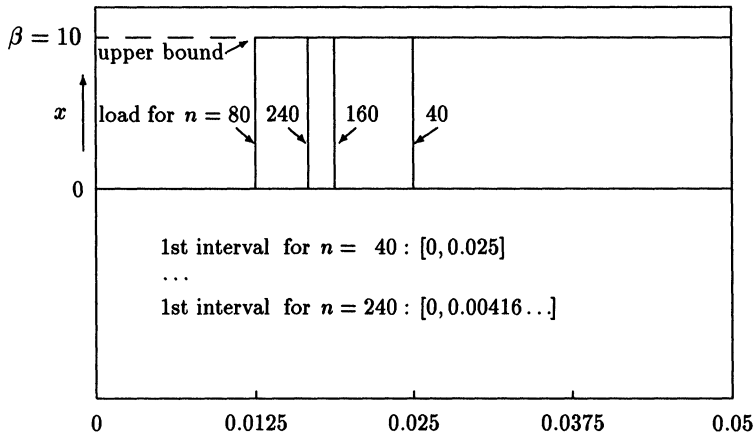

FIG. 4.3. *Load for different discretizations.*

to solve the quadratic programming problem (4.6). With its solution $y = y(x)$ and a Lagrange multiplier λ put

$$\tilde{K} := \left\{ z \in \mathbb{R}^{2n-2} \mid z_{2i-1} \text{ satisfies (4.8) for } i = 1, 2, \cdots, n-1 \right\},$$

$$z_{2i-1} \begin{cases} = 0, & \text{if } y_{2i-1} = g(\tfrac{i}{n}) \text{ and } \lambda_{2n-i} > 0, \\ \geq 0, & \text{if } y_{2i-1} = g(\tfrac{i}{n}) \text{ and } \lambda_{2n-i} = 0, \end{cases} \tag{4.8}$$

and solve the derived quadratic programming problem

$$\min_{z \in \tilde{K}} \tfrac{1}{2} z^T H z - \nabla_y l(y(x))^T z. \tag{4.9}$$

Then, under some technical assumptions (which we skip here), the following is proved in [33]: If p solves (4.9), then $\nabla b(x)^T p$ is a subgradient of f at x. Hence at every iteration we must solve two quadratic programming problems of dimension $2n-2$ to compute $f(x)$ and one $g \in \partial f(x)$. For these quadratic subproblems we use a code due to Powell. In our experiments we have set $\beta = 10$, $M = 5$, $g = -0.001$, and $n = 10$, 20, 40, 80, 160, 240. Further, we have incorporated the simple linear constraint $(1/n) \sum_{i=1}^{n} x_i = 5$ into the BTNC code itself. Figure 4.1 shows the deflection of the beam for $n = 10$ and for given $x = (5, 5, \cdots, 5)^T$ without optimizing; Fig. 4.2 gives the result for the "optimal" $x = (5, 10, 10, 0, \cdots, 0, 10, 10, 5)^T$ provided by BTNC.

We mention that these results differ substantially from those shown in [16] and [33]; a restart with BTNC from the data in that place proves that the results in [16] and [33] are not yet optimal. The "correctness" of our outcome is further confirmed by the BTNC solution for $n = 40$, 80, 160, and 240 (compare Fig. 4.3):

$$x = (x_1, x_2, \cdots, x_{11}, x_{12}, \cdots)^T = (0, 10, \cdots, 10, 0, \cdots)^T \in \mathbb{R}^{40},$$
$$x = (x_1, x_2, \cdots, x_{21}, x_{22}, \cdots)^T = (0, 10, \cdots, 10, 0, \cdots)^T \in \mathbb{R}^{80},$$
$$x = (x_1, x_2, x_3, x_4, \cdots, x_{43}, x_{44}, \cdots)^T = (0, 0, 0, 10, \cdots, 10, 0, \cdots)^T \in \mathbb{R}^{160},$$
$$x = (x_1, x_2, x_3, x_4, x_5, \cdots, x_{64}, x_{65}, \cdots)^T = (0, 0, 0, 0, 10, \cdots, 10, 0, \cdots)^T \in \mathbb{R}^{240}.$$

The stopping criterion was always $\varepsilon = 10^{-9}$.

Acknowledgment. We are indebted to the referees for their extremely careful reading of this paper and their valuable suggestions for improving the paper. We are also grateful to M. Grötschel, of the University of Augsburg, and U. Derigs, of the University of Bayreuth, for providing us with the data used in §4.2.

REFERENCES

[1] W. ACHTZIGER, *Nichtglatte Optimierung—Ein spezielles Problem bei der Durchführung von Versuchen,* Diplomarbeit, Universität Bayreuth, Bayreuth, Germany, 1989.

[2] B.M. BELL, *Global convergence of a semi-infinite optimization method,* Appl. Math. Optim., 21 (1990), pp. 69–88.

[3] J. CHARALAMBOUS AND A.R. CONN, *An efficient method to solve the minimax problem directly,* SIAM J. Numer. Anal., 15 (1978), pp. 162–187.

[4] F.H. CLARKE, *Nonsmooth Analysis and Optimization,* Wiley–Interscience, New York, 1983.

[5] J. CULLUM, W.E. DONATH, AND P. WOLFE, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices,* Math. Programming Stud., 3 (1975), pp. 35–55.

[6] V.F. DEMYANOV AND V.N. MALOZEMOV, *Introduction to Minimax,* John Wiley, New York, Toronto, 1974.

[7] U. DERIGS, *Another composite heuristic for solving Euclidean traveling salesman problems,* Preprint, Department of Mathematics, University of Maryland, College Park, MD, 1982.

[8] R.A. EL-ATTAR, M. VIDYASAGAR, AND S.R.K. DUTTA, *An algorithm for ℓ_1-norm minimization with application to nonlinear ℓ_1-approximation,* SIAM J. Numer. Anal., 16 (1979), pp. 70–86.

[9] YU.Y. ERMOLIEV, *Stochastic Programming Methods,* Nauka, Moscow, U.S.S.R., 1976.

[10] N. GAFFKE AND R. MATHAR, *Linear minimax estimation under ellipsoidal parameter space and related Bayes* L*-optimal design,* DFG-Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung," Report 42, 1988.

[11] J.L. GOFFIN, *Affine methods in nondifferentiable optimization,* Tech. Report, McGill University, Montreal, Québec, Canada, 1987.

[12] ———, Private communication.

[13] J.L. GOFFIN, A. HAURIE, AND J.P. VIAL, *Decomposition and Nondifferentiable Optimization with the Projective Algorithm,* GERAD, Faculty of Management, McGill University, Montreal, Québec, Canada, 1989.

[14] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *Geometric Algorithms and Combinatorial Optimization,* Springer-Verlag, Berlin, 1988.

[15] N. GUPTA, *A Higher than First Order Algorithm for Nonsmooth Constrained Optimization,* Ph.D. thesis, Department of Philosophy, Washington State University, Pullman, WA, 1985.

[16] J. HASLINGER AND P. NEITTAANMÄKI, *Finite Element Approximation for Optimal Shape Design,* John Wiley, Chichester, U.K., 1988.

[17] M. HELD AND R.M. KARP, *The traveling-salesman problem and minimum spanning trees: Part* 2, Math. Programming, 1 (1971), pp. 6–25.

[18] K.C. KIWIEL, *Methods of Descent for Nondifferentiable Optimization,* Springer-Verlag, Berlin, New York, 1985.

[19] ———, *A method for solving certain quadratic programming problems arising in nonsmooth optimization,* IMA J. Numer. Anal., 6 (1986), pp. 137–152.

[20] ———, *An ellipsoid trust region bundle method for nonsmooth convex optimization,* SIAM J. Control Optim., 27 (1989), pp. 737–757.

[21] ———, *Proximity control in bundle methods for convex nondifferentiable optimization,* Math. Programming, 46 (1990), pp. 105–122.

[22] C. LEMARÉCHAL, *A view of line searches,* in Optimization and Optimal Control, Lecture Notes in Control and Information Sciences, W. Oettli and J. Stoer, eds., Springer-Verlag, Berlin, Heidelberg, 1981.

[23] ———, *Extensions diverses des méthodes de gradient et applications,* Thèse d'Etat, Paris, 1980.

[24] ———, *Nondifferentiable optimization,* in Handbooks in Operations Research and Management Science, Vol. 1, Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, eds., North-Holland, Amsterdam, 1989.

[25] C. LEMARÉCHAL AND R. MIFFLIN, *A Set of Nonsmooth Optimization Test Problems,* in Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., Pergamon Press, Oxford, 1978, pp. 151–165.

[26] C. LEMARÉCHAL, J.-J. STRODIOT, AND A. BIHAIN, *On a bundle algorithm for nonsmooth opti-*

mization, in Nonlinear Programming 4, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, New York, 1981.

[27] C. LEMARÉCHAL AND M. BANCORA IMBERT, *Le Module* M1FC1, Tech. Report, Institut de Recherche d'Informatique et d'Automatique, Le Chesnay, 1985.

[28] K. MADSEN AND H. SCHJAER-JACOBSEN, *Linearly constrained minimax optimization*, Math. Programming, 14 (1978), pp. 208–223.

[29] M.M. MÄKELÄ, *Methods and algorithms for nonsmooth optimization*, Reports on Applied Mathematics and Computing, No. 2, Department of Mathematics, University of Jyväskylä, Jyväskylä, Finland, 1989.

[30] ———, *On the Methods of Nonsmooth Optimization*, Proc. 14th International Federation on Information Processing Conference on System Modelling and Optimization, Leipzig, Germany, 1989, to appear.

[31] R. MIFFLIN, *A modification and an extension of Lemaréchal's algorithm for nonsmooth optimization*, Math. Programming Stud., 17 (1982), pp. 77–90.

[32] ———, Private communication.

[33] J.V. OUTRATA, *On the numerical solution of a class of Stackelberg problems*, Zeitschrift für Operations Research, 4 (1990), pp. 255–278.

[34] B.T. POLJAK, *Subgradient methods: A survey of Soviet research*, in Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., Pergamon Press, Oxford, U.K., 1978, pp. 5–29.

[35] R.T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1972.

[36] K. SCHITTKOWSKI, *Heuristic reasoning in mathematical programming*, DFG-Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung", Report 209, 1990.

[37] H. SCHRAMM, *Eine Kombination von Bundle- und Trust-Region-Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*, Bayreuther Mathematische Schriften, Heft 30, Bayreuth, Germany, 1989.

[38] H. SCHRAMM AND J. ZOWE, *A combination of the bundle approach and the trust region concept*, in Mathematical Research, Vol. 45, Advances in Mathematical Optimization, J. Guddat et al., eds., Akademie-Verlag, Berlin, 1988, pp. 196–209.

[39] N.Z. SHOR, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, Heidelberg, 1985.

[40] G. SONNEVEND AND J. STOER, *Global ellipsoidal approximations and homotopy methods for solving convex analytic programs*, Appl. Math. Optim., 21 (1990), pp. 139–165.

[41] PH. WOLFE, *A method of conjugate subgradients for minimizing nondifferentiable convex functions*, Math. Programming Stud., 3 (1975), pp. 145–173.

[42] J. WOMERSLEY, *Numerical methods for structured problems in nonsmooth optimization*, Ph. D. thesis, Mathematics Department, University of Dundee, Dundee, Scotland, 1981.

[43] J. ZOWE, *Nondifferentiable optimization—A motivation and a short introduction into the subgradient- and the bundle-concept*, in NATO ASI Series, Vol. F15, Computational Mathematical Programming, K. Schittkowski, ed., Springer-Verlag, Berlin, Heidelberg, 1985.

[44] ———, *The* BT*-algorithm for minimizing a nonsmooth functional subject to linear constraints*, in Nonsmooth Optimization and Related Topics, F.H. Clarke, V.F. Demyanov, and F. Gianessi, eds., Plenum Press, New York, 1989, pp. 459–480.