Mathematical Programming 76 (1997) 393–410

Variable metric bundle methods: From conceptual to implementable forms

Claude Lemaréchal [*], Claudia Sagastizábal [1]

INRIA, B.P. 105, 78153 Le Chesnay, France

Received 7 November 1995; revised manuscript received 12 February 1996

Abstract

To minimize a convex function, we combine Moreau–Yosida regularizations, quasi-Newton matrices and bundling mechanisms. First we develop conceptual forms using "reversal" quasi-Newton formulae and we state their global and local convergence. Then, to produce implementable versions, we incorporate a bundle strategy together with a "curve-search". No convergence results are given for the implementable versions; however some numerical illustrations show their good behaviour even for large-scale problems.

Keywords: Convex optimization; Proximal point; Bundle method; Quasi-Newton
AMS Classification: Primary 65K05; Secondary 90C30; 52A41; 90C25

1. Introduction, notation, motivation

Consider the problem

$$\min_{x\in\mathbb{R}^N} f(x)\,, \tag{1}$$

where f is a convex function on $\mathbb{R}^N$, assumed finite valued for convenience. Associated with a symmetric positive-definite matrix M, we define the *Moreau–Yosida* regularization of f [25,31] by

$$F_M(x) := \min_{y\in\mathbb{R}^N}\{f(y) + \tfrac{1}{2}(y-x)^{\mathrm{T}}M(y-x)\}\,. \tag{2}$$

[*] Corresponding author. E-mail: claude.lemarechal@inria.fr
[1] E-mail: claudia.sagastizabal@inria.fr

PII S0025-5610(96)00053-6

We will also use a special notation for the optimal y in (2):

$$p_M(x) := \operatorname*{argmin}_{y\in\mathbb{R}^N}\{f(y) + \tfrac{1}{2}(y-x)^{\mathrm{T}}M(y-x)\}, \tag{3}$$

called the *proximal point* of x. It is known that F_M is convex and differentiable, that

$$\nabla F_M(x) = G := M(x - p_M(x)) \in \partial f(p_M(x)), \tag{4}$$

and that the minima of F_M coincide with those of f; furthermore ∇F_M is Lipschitzian. Then, admitting that some mechanism allows a fast computation of F_M and ∇F_M, the minimization of f via a quasi-Newton algorithm applied to F_M appears as very attractive. In fact, consider a conceptual algorithm whose nth iteration is as follows (see (AP1) in [2]).

Algorithmic Pattern 1 (AP1)

Step 1. For given x_n and $M_n \simeq \nabla^2 F_M(x_n)$, the solution $p_M(x_n)$ of (3) at $x = x_n$ is assumed available. Compute $\nabla F_M(x_n) = M(x_n - p(x_n))$.

Step 2. Make the quasi-Newton move

$$x_{n+1} = x_n - M_n^{-1}\nabla F_M(x_n) = x_n - M_n^{-1}M(x_n - p(x_n)), \tag{5}$$

possibly with some line-search to ensure global convergence.

Step 3. Compute $p_M(x_{n+1})$ from (3) and obtain $\nabla F_M(x_{n+1}) = M(x_{n+1} - p_M(x_{n+1}))$.

Step 4. Update M_n by a quasi-Newton formula using the pair $(\Delta x, \Delta G)$:

$$\Delta x = x_{n+1} - x_n, \qquad \Delta G = M(x_{n+1} - p_M(x_{n+1}) - x_n + p_M(x_n)).$$

Here and in the following, by *quasi-Newton formula using the pair* (u, v) we mean that the updated matrix in Step 4 satisfies the quasi-Newton equation

$$M_{n+1}u = v, \tag{6}$$

and we will write $M_{n+1} = \mathrm{qN}(M_n, u, v)$.

Remark that (AP1) with $M_n \equiv M$ results in $x_{n+1} = p_M(x_n)$, called the *proximal-point algorithm* [21,29], whose convergence properties are now rather well understood [9].

From a theoretical point of view, the conceptual algorithm (AP1) will converge globally (Lipschitz continuity of ∇F_M suffices for that, see [26]), and superlinearly when F_M happens to enjoy appropriate second-order smoothness [28]. Now, second-order smoothness of F_M, studied in [19], is not a simple thing. That is why we are not yet in a position to give conditions on f implying superlinear convergence of (AP1), albeit conceptual.

On the other hand, consider implementation aspects. A first drawback of (AP1) is the expensive computation of proximal points. As observed in [7,1,5], a bundle strategy can be applied to get round this difficulty. More precisely, assume that an *oracle* computes $f(y)$ and $g(y) \in \partial f(y)$ for any given y; after k iterations, the usual *cutting-plane* model of f is built:

$$\check{f}_k(y) := \max\{f(y_i) + g(y_i)^{\mathrm{T}}(y - y_i) : i = 1, 2, \dots, k\}, \tag{7}$$

where $(y_i, f(y_i), g(y_i) \in \partial f(y_i), i = 1, \dots, k)$ form the bundle. The whole idea of the method consists in replacing f by $\check{f}_k$ in (2):

Bundle Algorithmic Pattern (BAP)

Step 0. Given a symmetric positive-definite matrix M, set $n = k = 1$ and initialize $y_1 = x_1$.

Step 1. Having the nth *prox-center* x_n, compute the next iterate y_{k+1} as

$$y_{k+1} := \operatorname*{argmin}_y \{\check{f}_k(y) + \tfrac{1}{2}(y - x_n)^{\mathrm{T}} M (y - x_n)\}. \tag{8}$$

Step 2. With the help of a descent-test, decide whether to make
- either a *null-step*: just do nothing,
- or a *descent-step*: $x_{n+1} = y_{k+1}$ and increase n by 1.

Step 3. This completes the kth iteration: set $k = k + 1$ and update the model (7) by incorporating $(y_k, f(y_k), g(y_k))$ to the bundle. Loop to 1.

To the extent that $\check{f}_k$ approximates f, the optimal value in (8) (resp. y_{k+1}) can be viewed as an approximation of $F_M(x_n)$ (resp. $p_M(x_n)$). Indeed, if descent-tests were suppressed (so that x_n would remain fixed forever), the algorithm would eventually produce $F_M(x_n)$ and $p_M(x_n)$: this is Proposition 4.3 in [5] [2]. In the language of (AP1), a sequence of null-steps can be viewed as a subalgorithm approximating $p_M(x_n)$, while a descent-step mimics one iteration of the ordinary proximal-point algorithm. Incidentally, this interpretation shows that an appropriate descent-test is extremely important for efficiency, for approximations of F_M and convergence of x_n should groove together.

Suppose we want to merge the above implementation mechanism (BAP) with the quasi-Newton pattern (AP1). Knowing that $p_M(x_n)$ is approximated by y_{k+1}, (5) becomes in case of descent-step

$$x_{n+1} = x_n - M_n^{-1} M (x_n - y_{k+1}). \tag{9}$$

Here appears a second difficulty with (AP1): unless $M_n^{-1}M = I$, (9) is incompatible with the update formula $x_{n+1} = y_{k+1}$ in Step 2 of (BAP). Unfortunately, this update is crucial for the bundle mechanism to work, unless an approach as in [23] is adopted.

In this paper, we propose implementable algorithms issued from amending (AP1) + (BAP) according to the following principles:

Bundle principle: We update $x_{n+1} = y_{k+1}$ at each iteration of (AP1), i.e. at each descent-step of (BAP). For this, the metric M in (AP1) varies along the iterations. Indeed we do impose $M = M_n$ in (9).

Quasi-Newton principle: To update the metric M_n, we use a quasi-Newton formula aimed at yielding $M_{n+1} \simeq \nabla^2 F_{M_n}(x_n)$, as in (AP1). However, formula (6) will not

[2] We mention here that this result is true in finite dimension only; see the erratum in [27].

necessarily be based on the pair $(\Delta x, \Delta G)$. Rather, Section 3 proposes a *reversal* approach.

At first view, the idea may seem weird: there is no longer a fixed smooth function F_M to approximate the Hessian of; instead, there is rather a dog F_{M_n} running after its tail M_n; as a result, there is no longer a well-defined Hessian (of F_M) to be approximated via a series of quasi-Newton updates. However, we will see that the idea does have some merits: it results in a conceptual algorithm which converges superlinearly under reasonable assumptions on f; this is the subject of Sections 2–4. Finally, Section 5 addresses implementation issues; we propose a specific algorithm (whose convergence properties are not studied) and we present some numerical results.

2. Conceptual quasi-Newton proximal algorithms

As a first step, let us consider a quasi-Newton algorithm like (AP1), where x_n is updated according to the *Bundle principle* in a conceptual form:

$$x_{n+1} = p_{M_n}(x_n)\,, \quad \text{where } p_{M_n}(x_n) \text{ is computed exactly.}$$

Introducing a variable metric in (BAP) produces a variable proximal mapping in (3). For the sake of clarity, we use the following notation:

$$p_n(x) := \operatorname*{argmin}_{y\in\mathbb{R}^N}\{f(y) + \tfrac{1}{2}(y-x)^{\mathrm{T}} M_n (y-x)\}\,, \tag{10}$$

where M_n is symmetric positive definite. According to (4), the function F_{M_n} from (10) has the gradient

$$\nabla F_{M_n}(x_n) = G_n := M_n(x_n - p_n(x_n)) \in \partial f(p_n(x_n))\,. \tag{11}$$

Altogether, we consider the following class of conceptual *quasi-Newton proximal* algorithms.

Algorithmic Pattern 2 (AP2)

Step 0. The initial x_1 and M_1 are given. Set $n = 1$.

Step 1. Compute the proximal point $p_n(x_n)$ and set $x_{n+1} = p_n(x_n)$.

Step 2. Choose two vectors z and z'; compute the corresponding proximal points $p_n(z)$ and $p_n(z')$; set

$$u := z' - z\,, \qquad v := M_n(z' - p_n(z')) - M_n(z - p_n(z))\,.$$

Update M_n by some formula $M_{n+1} = \mathrm{qN}(M_n, u, v)$, using the pair (u, v).

Step 3. Increase n by 1 and loop to 1.

The algorithm is essentially of proximal type; the wording quasi-Newton refers only to the matrix-update strategy.

It is important to understand that the two reference points z and z' can be *chosen* in many ways. For instance, a standard quasi-Newton algorithm for minimizing directly f (rather than a regularization) would yield $M_{n+1} = \mathrm{qN}(M_n, \xi, \gamma)$, with $\xi := x_{n+1} - x_n$ and $\gamma := g(x_{n+1}) - g(x_n)$; but in general, the two vectors u and v defining our M_{n+1} can be a priori different from ξ and γ.

We establish now two abstract results which are independent of the way M_{n+1} is computed in (AP2). They concern convergence and speed of convergence, and they quantify the following intuition: taking "smaller" matrices M_n improves the convergence of the algorithm. Indeed, interpreting (10) as minimizing a "model" of $y \mapsto f(y)$, the best possible model is f itself, and this implies taking $M_n \equiv 0$.

Theorem 2.1. *Assume that the finite-valued convex function f has a nonempty bounded set of minima. Then the sequence $\{x_n\}$ generated by (AP2) is bounded. Moreover, if there is a constant $C \geqslant 0$ such that the eigenvalues of $\{M_n\}$ satisfy*

$$\lambda_{\max}(M_{n+1}) \leqslant \lambda_{\max}(M_n) + C, \quad \text{for all } n, \tag{12}$$

then any accumulation point of $\{x_n\}$ minimizes f.

The same result holds if we have $\mathrm{tr}(M_{n+1}) \leqslant \mathrm{tr}(M_n) + C$.

Proof. Repeated use of (12) gives

$$\lambda_{\max}(M_{n+1}) \leqslant nC + \lambda_{\max}(M_1)\,. \tag{13}$$

The series $\sum \lambda_{\max}^{-1}(M_n)$ is therefore divergent and the result follows from Theorem XV.4.2.3 in [10].

As for the trace relation, it implies likewise (13), because, for any $N \times N$ positive-definite matrix M, $\lambda_{\max}(M) \leqslant \mathrm{tr}(M) \leqslant N\lambda_{\max}(M)$ and this ends the proof. □

It is interesting to note that, if the matrices M_n are multiples of the identity, say $M_n = \mu_n I$ (a "poor man's" matrix), then $\{x_n\}$ is a minimizing sequence whenever $\sum \mu_n^{-1} = +\infty$, but with no assumption on $\mathrm{Argmin}\, f$ [9,12,5]. With general M_n, we do not know whether inf-compactness of f is still a necessary assumption for convergence. By contrast, the growth condition (14) below is necessary for superlinear convergence. The reason, as well as a counter-example, is given in [18]. Before giving a local convergence result, we state the following technical lemma, generalizing Lemma 4.3 in [2].

Lemma 2.2. *Let the convex function f have a unique minimum $\bar{x}$. Suppose there are $\varepsilon, \ell > 0$ such that $|y - \bar{x}| \leqslant \varepsilon$ implies*

$$f(y) \geqslant f(\bar{x}) + \ell|y - \bar{x}|^2\,. \tag{14}$$

Then, for every such y,

$$|g(y)|^2 \geqslant \ell(f(y) - f(\bar{x})), \quad \text{for any } g(y) \in \partial f(y)\,. \tag{15}$$

Proof. Write the subgradient inequality and apply Cauchy–Schwarz to obtain, for every $g(y) \in \partial f(y)$:

$$f(\bar{x}) \geqslant f(y) - |g(y)||\bar{x} - y|, \quad \text{or} \quad |g(y)||\bar{x} - y| \geqslant f(y) - f(\bar{x}) .$$

For y close enough to $\bar{x}$, the result follows easily from (14). □

Now we are in a position to give our superlinear convergence result.

Theorem 2.3. *Let f satisfy the assumptions of Lemma 2.2 and let (AP2) generate a sequence $\{x_n\}$ converging to $\bar{x}$. If $M_n(x_{n+1} - x_n) = o(|x_{n+1} - x_n|)$, then the convergence is superlinear: $|x_{n+1} - \bar{x}| = o(|x_n - \bar{x}|)$.*

Proof. Write (11); by construction, $M_n(x_n - x_{n+1}) =: G_n \in \partial f(x_{n+1})$. Then Lemma 2.2 gives $\ell|x_{n+1} - \bar{x}| \leqslant |G_n|$ for n large enough; hence, by assumption:

$$\ell|x_{n+1} - \bar{x}| = o(|x_{n+1} - x_n|) .$$

The rest is classical, based on the triangle inequality $|x_{n+1} - x_n| \leqslant |x_{n+1} - \bar{x}| + |\bar{x} - x_n|$. □

3. Reversal quasi-Newton formulae

Let us advance one more step in our way to deriving a reasonable implementable algorithm from (AP1) + (BAP). We address the problem of choosing z and z' in Step 2 of (AP2).

A first alternative, the most natural and standard one, is $z = x_n$, $z' = x_{n+1}$, which fixes $(u, v) = (\Delta x, \Delta G)$ as in Step 4 of (AP1). This, however, requires the computation of $p_n(x_{n+1})$, hence an extra resolution of (10). Furthermore, a serious difficulty arises when coming to numerical implementations; it concerns the property $v^{\mathrm{T}}u > 0$, crucial for quasi-Newton updates.

Fortunately, there is a way round these two pitfalls, based on a tricky choice of z and z'. In fact, consider the situation from the opposite point of view. Suppose we fix first v, that is to say, we take (sub)gradients at two different points; can we find the corresponding u? The answer lies in the following result, which states that the proximal operator has an explicit inverse:

Theorem 3.1. *Take $y \in \mathbb{R}^N$ and $g \in \partial f(y)$. Then y is a posteriori the solution of (10) with x replaced by $y + M_n^{-1}g$. In other words*

$$y = p_n(y + M_n^{-1}g), \quad \textit{for any } y \in \mathbb{R}^N \textit{ and } g \in \partial f(y) .$$

Proof. Write the optimality condition for problem (10). □

Thus, when we have an arbitrary point $y \in \mathbb{R}^N$ (for example x_n or x_{n+1}) and an arbitrary subgradient of f at y (for example $g(x_n)$ or $g(x_{n+1})$), we also have a proximal point for free, and we therefore have a gradient ∇F_{M_n} for free as well. Said otherwise: given a point z, it is difficult to compute the gradient $\nabla F_{M_n}(z)$; but given a vector G which is a subgradient of f, Theorem 3.1 provides us immediately with a point z such that $G = \nabla F_{M_n}(z)$ - a "reversal" operation.

Back to our quasi-Newton context, instead of making the standard choice (fix u, then compute v), we will *first* fix the difference of gradients v and then we will use the reversal operation to deduce the corresponding u. More precisely, take v as *any* difference of subgradients of f at x_n and x_{n+1}. Then, recalling

$$\xi = x_{n+1} - x_n \,, \tag{16}$$

we directly have

$$u = \xi + M_n^{-1} v \,. \tag{17}$$

This results in the quasi-Newton update $M_{n+1} = \mathrm{qN}(M_n, \xi + M_n^{-1}v, v)$ which has several advantages.

First of all, the formula is explicit and requires no extra prox-computation.

Also, a wide variety is available for v. Indeed, when the nth iteration of an algorithm is completed, we always have on hand $g(x_n) \in \partial f(x_n)$ and $g(x_{n+1}) \in \partial f(x_{n+1})$; but we also have from (11) the subgradients $G_{n-1} \in \partial f(x_n)$ and $G_n \in \partial f(x_{n+1})$. Thus, we can take

$$v := \gamma = g(x_{n+1}) - g(x_n) \,, \tag{18}$$

or

$$v := G_n - G_{n-1} \,, \tag{19}$$

and indeed infinitely many other choices using convex combinations.

All of these choices have their respective qualities. For example, we will see in Section 4 that (19) allows superlinear convergence (which is only conceptual, because G_{n-1} and G_n cannot be computed exactly: at the implementation stage, f in (11) has to be replaced by $\check{f}_k$ from (7)). On the other hand, (18) is useful for ensuring positive definiteness of the matrices. In fact, the property $v^{\mathrm{T}}u > 0$ becomes here

$$v^{\mathrm{T}}u = v^{\mathrm{T}}\xi + v^{\mathrm{T}}M_n^{-1}v > 0 \,. \tag{20}$$

With formula (18), we have $v^{\mathrm{T}}\xi = \gamma^{\mathrm{T}}\xi$, which is easily made positive, via a Wolfe-type line-search [14]. We will come back to this property in Section 5.

4. Convergence properties of various conceptual quasi-Newton updates

In this section, we consider the algorithmic pattern (AP2) with z and z' such that $p_n(z) = x_n$, $p_n(z') = x_{n+1}$. In other words, we take ξ and u as described in (16), (17). There is more freedom for the choice of v, recall (18), (19). We apply the abstract convergence results of Section 2 to various reversal qN-formulae.

4.1. Symmetric rank one

Let us start with the symmetric rank-one formula (see [6] for example), which does not have a good reputation, because it is unstable: it involves a denominator which may well vanish. Thanks to (17) such is not the case when the updating pair (u, v) corresponds to a Moreau–Yosida regularization:

Theorem 4.1. *Assume M_n is positive-definite. The reversal symmetric rank-one update is*

$$M_{n+1} = M_n - \frac{M_n \xi \xi^{\mathrm{T}} M_n}{v^{\mathrm{T}}\xi + \xi^{\mathrm{T}} M_n \xi}, \tag{21}$$

where v is given by (18) or (19). If $v^{\mathrm{T}}\xi > 0$, then M_{n+1} is well-defined and positive definite. In these circumstances, the resulting sequence $\{x_n\}$ in Algorithm (AP2) is minimizing if f has a bounded set of minima.

Proof. The symmetric rank-one update $\mathrm{qN_{SR1}}(M_n, u, v)$ has the form $M_{n+1} = M_n + \alpha\zeta\zeta^{\mathrm{T}}$, where $(\zeta, \alpha) \in (\mathbb{R}^N, \pm 1)$ is the unique pair such that the quasi-Newton equation (6) is satisfied. Working out the calculations and taking (17) into account, we obtain $\zeta = M_n\xi$ and $\alpha = -1/\xi^{\mathrm{T}} M_n u$; (21) holds.

Now suppose $v^{\mathrm{T}}\xi > 0$. Use $\det(I + \alpha a b^{\mathrm{T}}) = 1 + \alpha a^{\mathrm{T}} b$ in (21):

$$\det M_{n+1} = \det M_n \left(1 - \frac{\xi^{\mathrm{T}} M_n \xi}{v^{\mathrm{T}}\xi + \xi^{\mathrm{T}} M_n \xi}\right) = \det M_n \frac{v^{\mathrm{T}}\xi}{v^{\mathrm{T}}\xi + \xi^{\mathrm{T}} M_n \xi},$$

which is positive if M_n is positive definite. We conclude that M_{n+1} has an even number of negative eigenvalues, if any. According to the Interlacing Eigenvalue Theorem (see Lemma 7.4 in [6], for instance), the rank-one formula (21) gives

$$\lambda_{\min}(M_{n+1}) \leqslant \lambda_{\min}(M_n) \leqslant \ldots \leqslant \lambda_{\max}(M_{n+1}) \leqslant \lambda_{\max}(M_n).$$

Since $\lambda_{\min}(M_n) > 0$, the number of non-positive eigenvalues of M_{n+1} is at most one; this number is therefore 0: M_{n+1} is positive definite.

Finally, the above interlacing property allows the use of Theorem 2.1 with $C = 0$ in (12). $\square$

4.2. A poor man's quasi-Newton formula

Another possible formula consists in restricting M_n to be a multiple of the identity $M_n = \mu_n I$; we will call the resulting algorithmic pattern “poor man's (AP2)”. When M_{n+1} is so restricted, the quasi-Newton equation (6) is impossible to solve; rather the update factor μ_{n+1} solves the variational quasi-Newton problem $\min_\mu \frac{1}{2}|v/\mu - u|^2$. Because f is convex, $v^{\mathrm{T}} u \geqslant 0$. Actually, this variational problem has a (positive) solution only if $v^{\mathrm{T}} u > 0$ and this holds, for example, when f is strictly convex. Then $\mu_{n+1} = |v|^2 / v^{\mathrm{T}} u$, so, using (17), we have

$$\frac{1}{\mu_{n+1}} = \frac{v^T\xi}{|v|^2} + \frac{v^T M_n^{-1} v}{|v|^2} = \frac{v^T\xi}{|v|^2} + \frac{1}{\mu_n}. \tag{22}$$

Various convergence results follow.

Theorem 4.2. *Assume f is a strictly convex function. Then the poor man's (AP2) generates a sequence $\{x_n\}$ which is minimizing, and which converges to the minimum of f, if it exists.*

Proof. From (22), $\mu_{n+1} \leqslant \mu_n$; then Corollary 2.3 in [5] applies. □

We also have some superlinear convergence results. We start with a validation of our approach in the smooth case:

Theorem 4.3. *Make the assumptions of Lemma 2.2. In addition, let f be strictly convex with a locally Lipschitz continuous gradient. Then the poor man's (AP2) is superlinearly convergent.*

Proof. We already know that $\{x_n\}$ converges to the minimum of f. Knowing that $v = \nabla f(x_{n+1}) - \nabla f(x_n) = \gamma$, the local Lipschitz property of ∇f implies $|v|^2 \leqslant L v^T \xi$ (see Theorem X.4.2.2 in [10], for example). Then (22) shows that $1/\mu_n \to +\infty$; apply Theorem 2.3 to conclude. □

With the assumptions of Theorem 4.3, the choice (18) or (19) is irrelevant: $G_{n-1} = g(x_n) = \nabla f(x_n)$. When f is not differentiable, we can establish superlinear convergence only with the specific choice (19) for v. Recall that f is strongly convex with modulus $c > 0$ if for any y and x

$$f(y) \geqslant f(x) + g(x)^T(y-x) + \tfrac{1}{2}c|y-x|^2, \quad \text{for all } g(x) \in \partial f(x). \tag{23}$$

Theorem 4.4. *Let f be strongly convex with modulus c. Then the reversal poor man's (AP2) with v from (19) generates a sequence $\{f(x_n)\}$ which is 2-step superlinearly convergent:*

$$f(x_{n+1}) - \bar{f} = o(f(x_{n-1}) - \bar{f}),$$

where $\bar{f}$ denotes the optimal value of f.

Proof. From the assumptions, f has a unique minimizer and x_n converges to it. The positive decreasing sequence $\{\mu_n\}$ has a limit $\bar{\mu} \geqslant 0$. If $\bar{\mu} = 0$, then $\mu_n(x_{n+1} - x_n) = o(|x_{n+1} - x_n|)$ and Theorem 2.3 ensures the superlinear convergence. If $\bar{\mu} > 0$, (22) shows that $0 \leqslant v^T\xi/|v|^2 =: \Delta_n$ must tend to 0.

We proceed to bound Δ_n from below. Write $|v|^2 = |G_n|^2 + |G_{n-1}|^2 - 2G_n^T G_{n-1}$ and introduce the notation $f_n := f(x_n)$. Using (11), the appropriate subgradient inequalities and $\{\mu_n\}$ decreasing, it holds

$$|G_n|^2 = -\mu_n G_n^{\mathrm{T}}(x_{n+1} - x_n) \leqslant \mu_0(f_n - f_{n+1}),$$
$$|G_{n-1}|^2 = -\mu_{n-1} G_{n-1}^{\mathrm{T}}(x_n - x_{n-1}) \leqslant \mu_0(f_{n-1} - f_n)$$

as well as

$$-2G_n^{\mathrm{T}} G_{n-1} = 2\mu_n(x_{n+1} - x_n)^{\mathrm{T}} G_{n-1} \leqslant 2\mu_0(f_{n+1} - f_n) \leqslant 0$$

(remember $f_{n+1} = f(p_n(x_n)) \leqslant f(x_n)$). Altogether, we get

$$|v|^2 = |G_n - G_{n-1}|^2 \leqslant \mu_0(f_n - f_{n+1} + f_{n-1} - f_n) = \mu_0(f_{n-1} - f_{n+1}). \tag{24}$$

On the other hand, apply (23) twice, exchanging $y = x_{n+1}$ and $x = x_n$, to obtain directly $v^{\mathrm{T}}\xi \geqslant c|\xi|^2$. Now, strong convexity allows us to apply Lemma 2.2 with $\ell = \frac{1}{2}c$ to $G_n = -\mu_n \xi \in \partial f(x_{n+1})$ to obtain

$$v^{\mathrm{T}}\xi \geqslant c|\xi|^2 \geqslant \frac{c^2}{2\mu_n^2}(f_{n+1} - \bar{f}) \geqslant \frac{c^2}{2\mu_0^2}(f_{n+1} - \bar{f}). \tag{25}$$

From (24) and (25) we conclude

$$0 \leftarrow \Delta_n = \frac{v^{\mathrm{T}}\xi}{|v|^2} \geqslant K \frac{f_{n+1} - \bar{f}}{f_{n-1} - f_{n+1}} \geqslant 0,$$

with $K := c^2/2\mu_0^3$. Then, for any $\varepsilon > 0$, there exists an index n_ε such that

$$f_{n+1} - \bar{f} \leqslant \varepsilon(f_{n-1} - f_{n+1}) = \varepsilon(f_{n-1} - \bar{f}) - \varepsilon(f_{n+1} - \bar{f}), \quad \text{for all } n \geqslant n_\varepsilon.$$

Equivalently,

$$f_{n+1} - \bar{f} \leqslant \frac{\varepsilon}{1+\varepsilon}(f_{n-1} - \bar{f}), \quad \text{for all } n \geqslant n_\varepsilon,$$

and this ends the proof. □

5. Implementable forms

To become implementable, (AP2) must be combined with (BAP), whose role is to approximate $p_n(x_n)$, via a bundling subalgorithm in which f itself is approximated by polyhedral functions $\check{f}_k$. However, we propose an implementation which does not limit the role of (BAP) to this "approximal" computation. In fact, once M_n is fixed, the subalgorithm may also modify the weight of the quadratic penalty in (10), and this for two reasons:

(i) As already mentioned, it is important to guarantee (20). However, the next iterate $x_{n+1} = y_{k+1}$ coming out of (8) may be so close to x_n that no subgradient of f at x_{n+1} yields the required inequality (20). This will happen when M_n is a "big" matrix, or also when f is affine near x_n. Before making a descent-step, we therefore test (20); if it is not satisfied, we *extrapolate*, dividing M_n by a coefficient $t > 1$.

(ii) The results in Section 4 suggest that our quasi-Newton updates are only able to "decrease" M_n; at least, this is clearly the case with (21) and (22). This decrease is a good idea for the conceptual algorithm: after all, (10) shows that best results are obtained with $M_n = 0$! However, since $p_n(x_n)$ is not computed exactly, we may need "big enough" matrices M_n: the case $M = 0$ transforms (BAP) into the pure cutting-plane algorithm [4,11], threatened by Nemirovski's disastrous counter-example (see for example §XV.1.1 of [10]). For this reason, the bundling subalgorithm may also *interpolate*, dividing M_n by a coefficient $t < 1$ when suitable.

Thus, at each iteration of (AP2) + (BAP), (10) is replaced by

$$\check{p}(t) := \operatorname*{argmin}_{y\in\mathbb{R}^n}\left\{\check{f}_k(y) + \frac{1}{2t}(y - x_n)^{\mathrm{T}}M_n(y - x_n)\right\}. \tag{26}$$

The next candidate is then

$$y_{k+1} = \check{p}(t) = x_n - tM_n^{-1}\check{G}(t), \tag{27}$$

where, just as in (4) or (11),

$$\check{G}(t) := \frac{M_n}{t}(x_n - \check{p}(t)) \in \partial\check{f}_k(\check{p}(t)). \tag{28}$$

Observe that $\check{p}(t)$ describes a curve when the "stepsize" $t > 0$ varies. In effect, we are performing a "curve-search" (comparable to a line-search), analogous to the trust-region technique [24], and also outlined in [12,30]. This technique allows us to distinguish two different aspects of the "ideal" metric: a *shape* M_n, taken care of by the quasi-Newton updates, and a *size* t or t_n, managed by the curve-search. With this distinction, the *Bundle Principle* is preserved: when the curve-search is finished with a descent-stepsize t_n, the matrix update becomes $M_{n+1} = \mathrm{qN}(M_n/t_n, u, v)$.

Now, we have to specify the rules to compute t within the curve-search. To understand our technique, some familiarity with the bundle jargon is advisable; the reader is referred to Chapter XV of [10] for full explanations. The resolution of (26) yields a positive *nominal decrease*

$$\delta(t) := f(x_n) - \check{f}_k(\check{p}(t)) + \tfrac{1}{2}\check{G}(t)^{\mathrm{T}}(\check{p}(t) - x_n)$$

and a nonnegative *linearization error*

$$e(t) := f(x_n) - f(\check{p}(t)) + g(\check{p}(t))^{\mathrm{T}}(\check{p}(t) - x_n). \tag{29}$$

Unless x_n solves (1), $\check{G}(t) \neq 0$. Using $f(x_n) \geqslant \check{f}_k(x_n)$ and (28), there also holds

$$\delta(t) \geqslant \tfrac{1}{2}\check{G}(t)^{\mathrm{T}}(x_n - \check{p}(t)) = \tfrac{1}{2}t\check{G}(t)^{\mathrm{T}}M_n^{-1}\check{G}(t). \tag{30}$$

The following properties will be useful for our convergence analysis.

Lemma 5.1. *Consider (26) as parameterized by t. Then the following holds:*

(i) $\check{G}(\cdot)^{\mathrm{T}}M_n^{-1}\check{G}(\cdot)$ *is nonincreasing.*

(ii) *$\delta(\cdot)$ and $\check{p}(\cdot)$ are continuous at any $t > 0$.*
(iii) *When $t \downarrow 0$, $\check{p}(t) \to x_n$ and $|\check{p}(t) - x_n| = O(t)$.*

Proof. (i) For a given $t > 0$, we dualize (26); applying for example Fenchel's duality theorem, $\check{G}(t)$ solves

$$\min_{G \in \mathbb{R}^N} \left\{ \tfrac{1}{2} G^{\mathrm{T}} M_n^{-1} G + \frac{1}{t}(\check{f}_k^*(G) - G^{\mathrm{T}} x_n) \right\}.$$

Interpret $1/t$ as a Lagrange multiplier associated to the constraint in

$$\begin{cases} \min \tfrac{1}{2} G^{\mathrm{T}} M_n^{-1} G, \\ \check{f}_k^*(G) - G^{\mathrm{T}} x_n \leqslant \theta. \end{cases}$$

When θ increases, the multiplier $1/t$ decreases and the objective function decreases. Altogether, (i) follows.

(ii) Straightforward, observing that $\check{p}(t)$ is the unique optimal solution and $f(x_n) - \delta(t)$ is the optimal value in the parameterized problem (26).

(iii) When $t \downarrow 0$, the continuity of $\check{p}(\cdot)$ still holds: indeed, $\check{p}(t)$ is not changed if the objective function in (26) is multiplied by $t > 0$. Then, $\check{p}(t) \to \check{p}(0) = x_n$. Finally, the finite-valued convex function $\check{f}_k$ is locally Lipschitzian: in (27), $\check{G}(\cdot)$ is bounded and the rate of convergence follows. □

The core of any bundle method is the descent-test in Step 2 of (BAP). A descent-step is allowed when

$$f(\check{p}(t)) \leqslant f(x_n) - m_1 \delta(t), \tag{31}$$

where $m_1 \in]0, 1[$ is an Armijo-like tolerance. Keeping this in mind, we can outline the curve-search algorithm.

Extrapolations. Increasing t is done when (31) holds, and aims at satisfying (20) as well. If v is chosen as in (18), we want $g(\check{p}(t))^{\mathrm{T}}(\check{p}(t) - x_n) > g(x_n)^{\mathrm{T}}(\check{p}(t) - x_n)$; we are essentially in the framework of Wolfe's line-search. However, the special form of $\check{p}(t)$ brings a slight complication. Because $\check{f}_k$ is polyhedral, the following properties hold (see for example Remark XV.4.2.6 in [10], see also [13]):

- Either $\check{f}_k$ is bounded from below. Then it attains its minimum and, when t increases, $\check{p}(t)$ stops at such a minimum. In such a case, extrapolations may become helpless. We therefore check whether $\check{p}(t)$ becomes constant; if yes, the curve-search exits, producing a "cutting-plane-step": $\check{p}(t)$ minimizes $\check{f}_k$.
- Or $\check{f}_k$ is unbounded from below. Then we have the useful property

$$\text{there is } c > 0 \text{ such that } |G| \geqslant c, \text{ for all } G \in \partial \check{f}_k(p) \text{ and } p \in \mathbb{R}^N; \tag{32}$$

thus, (28) shows that extrapolations will drive $\check{p}(t)$ to infinity, as in a usual line-search.

Interpolations. Reducing t is quite a delicate business. We give a possibility, mentioning here that other techniques might be more efficient; what we lack is to fully understand all the intricacies of the bundling mechanism. First of all, interpolation (increase of the quadratic penalty in (26)) is advisable only when the bundling mechanism comes into play, i.e. when the descent-test is not satisfied. In fact, when (31) is satisfied, it is certainly a bad idea to increase the penalty, thereby decreasing the target $\delta(t)$; one should rather be optimistic and do the contrary. Second, the next cutting-plane model $\check{f}_{k+1}$ will incorporate the new bundle element $\{\check{p}(t), f(\check{p}(t)), g(\check{p}(t))\}$; it is known by empirical observations that this new element has a limited influence if $e(t)$ of (29) is large (then the next candidate y_{k+1} will not be much different from y_k). Accordingly, we propose to accept a null-step only when

$$e(t) \leqslant m_3\delta(t)\,, \tag{33}$$

m_3 being a positive tolerance.

We are now in a position to state the implementable algorithm.

Reversal quasi-Newton Bundle Algorithm (RQB)

Step 0. The initial x_1 and M_1 are given, as well as curve-search tolerances $0 < m_1 < m_2 < 1$ and $m_3 > 0$. Define the initial cutting-plane model $\check{f}_1$ according to (7) and set $k = n = 1$.

Step 1. Set $t = 1$ and call the curve-search subalgorithm (CS) below, which answers: a new $t > 0$, a candidate $\check{p}(t)$ with its associated $\check{G}(t)$, and a message indicating:
- either null-step, when (31) does not hold but (33) holds;
- or descent-step, when (31) holds, as well as

$$g(\check{p}(t))^{\mathrm{T}}(\check{p}(t) - x_n) \geqslant -m_2\delta(t)\,; \tag{34}$$

- or cutting-plane-step, when (31) holds and $\check{p}(t)$ minimizes $\check{f}_k$.

Step 2. In case of null-step, go to Step 3. Otherwise update $x_{n+1} = \check{p}(t)$ and set $G_n := \check{G}(t)$. In case of descent-step, set $\xi := x_{n+1} - x_n$. Compute v by a formula like (18) or (19) and update $M_{n+1} = \mathrm{qN}(M_n/t, \xi + tM_n^{-1}v, v)$. Replace n by $n+1$.

Step 3. Update the cutting-plane model $\check{f}_{k+1}$ of (7). Replace k by $k+1$ and loop to Step 1.

The curve-search algorithm (CS) will follow the usual principles of a Wolfe-type line-search, see [14] for example. The main differences are:
- The trial points $\check{p}(t)$ are computed via (26), instead of having the simple form $\check{p}(t) = x_n + td$.
- The possible exit by "null-step" will prevent $t \downarrow 0$.
- The additional exit by "cutting-plane-step" copes with the case of a bounded $\check{p}(t)$ for $t \to +\infty$.

Remark 5.2. After (26) is solved, we obtain in addition to $\check{G}(t)$ of (28):

$$\check{e}(t) := f(x_n) - \check{f}_k(\check{p}(t)) + \check{G}(t)^{\mathrm{T}}(\check{p}(t) - x_n)\,.$$

With this notation, we have $\check{G}(t) \in \partial_{\check{e}(t)} f(x_n)$:

$$f(y) \geqslant f(x_n) + \check{G}(t)^{\mathrm{T}}(y - x_n) - \check{e}(t), \quad \text{for all } y \in \mathbb{R}^N.$$

The stopping criterion can therefore be performed as usual: having two tolerances $\varepsilon > 0$ and $\eta > 0$,

$$\text{stop if } |\check{G}(t)| \leqslant \eta \text{ and } \check{e}(t) \leqslant \varepsilon. \tag{35}$$

Then we can write

$$f(y) \geqslant f(x_n) - \varepsilon - \eta|y - x_n|, \quad \text{for all } y \in \mathbb{R}^N.$$

Besides, the test for a cutting-plane-step checks the magnitude of $\check{G}(t)$. More precisely, the test for a cutting-plane-step will be

$$|\check{G}(t)| \leqslant \eta, \quad \text{or} \quad \check{G}(t)^{\mathrm{T}}(\check{p}(t) - x_n) \geqslant -m_4\check{e}(t)\,, \tag{36}$$

for some positive coefficient m_4 chosen at Step 0 of (RQB). □

The general idea of the curve-search is: given $t > 0$, we perform the descent-test; if it is satisfied (resp. not satisfied), t is declared a t_{L} (t-left, too small) (resp. a t_{R} (t-right, too large)). The algorithm iteratively reduces the bracketing interval $]t_{\mathrm{L}}, t_{\mathrm{R}}[$ by successive interpolations, possibly preceded by some extrapolations.

Curve-search Algorithm (CS)

Step 0. Set $t_{\mathrm{L}} = 0$, $t_{\mathrm{R}} = +\infty$.

Step 1. For the given current $t > 0$, solve (26) to obtain $\check{p}(t)$, $\check{G}(t)$, $\delta(t)$, $\check{e}(t)$. Perform the stopping criterion (35).

Step 2. If (31) is satisfied, set $t_{\mathrm{L}} = t$ and go to Step 4.

Step 3. (no descent obtained)
Set $t_{\mathrm{R}} = t$. If $t_{\mathrm{L}} = 0$ and (33) is satisfied, declare "null-step" and return. Else go to Step 6.

Step 4. (descent obtained)
If (34) is satisfied, declare "descent-step" and return.

Step 5. If $t_{\mathrm{R}} = +\infty$ and (36) holds, declare "cutting-plane-step" and return.

Step 6. Compute a new t, either by extrapolation ($t_{\mathrm{R}} = +\infty$) or by interpolation ($t_{\mathrm{R}} < +\infty$). Loop to Step 1.

Note how the stopping criterion has been placed *within* (CS), after each resolution of (26) in Step 1. Note also that, to implement (19) for v in Step 2 of (RQB), we are bound to replace the unknown G_n by $\check{G}_n := \check{G}(t)$ (here t is the final value answered by (CS)).

Proving convergence of (RQB) + (CS) is not so easy. However, (CS) is consistent: the exit eventually occurs at Step 3, Step 4 or Step 5 (or at Step 1, due to the stopping criterion). To prove this, we introduce the concept of "safeguard-reduction property":

Infinitely many extrapolations lead t_{L} to infinity.

Infinitely many interpolations lead $t_{\mathrm{R}} - t_{\mathrm{L}}$ to zero.

Theorem 5.3. *Let f be a finite-valued convex function and suppose that x_n does not minimize f. If the safeguard-reduction property holds, then Algorithm (CS) satisfies the following alternative: either $f(\check{p}(t)) \to -\infty$, or a descent-, a null- or a cutting-plane-step is eventually returned.*

Proof. Suppose first that $t_{\mathrm{R}} \equiv +\infty$ (Step 3 never visited); then $t \to +\infty$ and there are two cases. If the polyhedral $\check{f}_k$ is unbounded from below, $|\check{G}(t)|$ satisfies (32). Use (30) to obtain with (28)

$$\delta(t) \geqslant \tfrac{1}{2}t\check{G}(t)^{\mathrm{T}}M_n^{-1}\check{G}(t) \geqslant \frac{c^2}{2\lambda_{\max}(M_n)}t \to +\infty, \quad \text{when } t \to +\infty.$$

This, together with the descent-test (31), shows that $f(\check{p}(t)) \to -\infty$. Otherwise, $\check{f}_k$ has a minimum p_{CP} which is attained by $\check{p}(t)$ for t large enough; then $\check{G}(t) = (p_{\mathrm{CP}} - x_n)/t \to 0$ and a cutting-plane-step is eventually produced in Step 5.

Suppose now that some finite t_{R} is produced so that $\{t_{\mathrm{L}}\}$ and $\{t_{\mathrm{R}}\}$ form two adjacent sequences with a common limit t^*. Then there are again two cases. When $t^* > 0$, the descent-test (31) gives

$$f(\check{p}(t_{\mathrm{R}})) > f(x_n) - m_1\delta(t_{\mathrm{R}}) \quad \text{and} \quad f(\check{p}(t_{\mathrm{L}})) \leqslant f(x_n) - m_1\delta(t_{\mathrm{L}}).$$

Passing to the limit,

$$f(\check{p}(t^*)) = f(x_n) - m_1\delta(t^*). \tag{37}$$

On the other hand, t_{L} does not satisfy (34): $g(\check{p}(t_{\mathrm{L}}))^{\mathrm{T}}(\check{p}(t_{\mathrm{L}}) - x_n) < -m_2\delta(t_{\mathrm{L}})$. So, using $g(\check{p}(t_{\mathrm{L}})) \in \partial f(\check{p}(t_{\mathrm{L}}))$, we have $f(\check{p}(t_{\mathrm{L}})) - f(x_n) < -m_2\delta(t_{\mathrm{L}})$. Passing to the limit, $f(\check{p}(t^*)) - f(x_n) \leqslant -m_2\delta(t^*)$. With (37), this yields the contradiction $0 \leqslant (m_1 - m_2)\delta(t^*) < 0$.

In the last case, $t_{\mathrm{R}} \downarrow t^* = 0$. Use (30) and Lemma 5.1 (i) to obtain

$$\delta(t) \geqslant \tfrac{1}{2}t\check{G}(t)^{\mathrm{T}}M_n^{-1}\check{G}(t) \geqslant \tfrac{1}{2}t\check{G}(1)^{\mathrm{T}}M_n^{-1}\check{G}(1). \tag{38}$$

Now consider the normalized direction $d(t) := (\check{p}(t) - x_n)/|\check{p}(t) - x_n|$. Use convexity in the definition (29) of $e(t)$ and divide by $|\check{p}(t) - x_n| > 0$, to obtain

$$\frac{e(t)}{|\check{p}(t) - x_n|} \leqslant g(\check{p}(t))^{\mathrm{T}}d(t) - f'(x_n; d(t)).$$

Let d be a cluster-point of $\{d(t)\}$. Then $f'(x_n; d(t)) \to f'(x_n; d)$ (the finite-valued convex function $f'(x_n; \cdot)$ is continuous). Also, $g(\check{p}(t))^{\mathrm{T}}d(t) \to f'(x_n; d)$ (the convex function f is semi-smooth [22]). Hence,

$$\frac{e(t)}{|\check{p}(t) - x_n|} \to 0,$$

so, in view of Lemma 5.1 (iii), $e(t) = o(t)$. Recalling (38), we see that the null-test (33) in Step 4 must cut the interpolations. □

Table 1
Results on standard test problems

	MAXQUAD	TR48	TSP442	TSP1173	TSP3038
(RQB)	86	216	210	276	790
Brännlund	140	350	–	–	–
Kiwiel	41	170	–	–	–
Level	88	105	–	–	–
Dual bundle	82	385	1398	413	$+\infty$
BT	56	169	233	140	4142

Numerical illustrations. To finish the paper we report some experiments with the combination (RQB) + (CS) + poor man's quasi-Newton formula: M_n has the form $M_n := \mu_n I$, and μ_n is given by (22). Our choice of v is based on the following observation: even though fast convergence relies on a decrease of μ_n, we observed that μ_n has a tendency to stall. In fact, this can be seen from (22): usually $\xi = x_{n+1} - x_n \to 0$, but v neither in (18) nor in (19) has any reason to be small; hence the increase in $1/\mu_n$ may become negligible. Accordingly, we choose among the four values of v

$$G_n - G_{n-1}, \qquad G_n - g(x_n), \qquad g(x_{n+1}) - G_{n-1}, \qquad g(x_{n+1}) - g(x_n),$$

the one giving the smallest μ_{n+1}.

Table 1 shows our results on standard test problems: MAXQUAD and TR48 are described in [15], TSP's can be obtained by anonymous ftp at the address ftp://softlib.cs.rice.edu/pub/tsplib[3]. We give the number of oracle calls to reach 10^{-4} relative accuracy. When available we also give the corresponding results obtained by the algorithms described in [3] (Brännlund), [12] (Kiwiel), [16] (Level), [20] (Dual bundle) and [30] (BT).

Note in particular that TSP3038, with its 3038 variables, is quite a challenge for nonsmooth optimization. Naturally, storing all the $g(y_i) \in \mathbb{R}^{3038}$, $i = 1, \ldots, k$ becomes prohibitive when k goes beyond 10^3: a special device is needed to limit storage. We adopted a fairly standard technique: when k reaches a maximal value $k_{\max}$, one element is deleted from the bundle, thus making space for the new element. We delete the element having the largest e from (29), among those that are not active in the value of the current $\check{f}_k(\check{p}(t))$. The implementations reported in Table 1 used $k_{\max} = 500$.

[3] Beware that these versions of TSP's are not those previously used in the nonsmooth literature, which were very sensitive to roundoff errors.

6. Conclusion

Starting from theoretical properties of the Moreau–Yosida regularization, we have developed a class of algorithms combining quasi-Newton updates with the bundling mechanism. We have introduced a reversal formula, well-suited to the implicit character of the Moreau–Yosida regularization. The resulting conceptual algorithms enjoy appropriate global and local convergence properties. In fact, they bring interesting answers to the long-lasting question of managing the quadratic term in a proximal-bundle method. Besides, they have convenient implementable forms whose numerical behaviour compares well with other existing methods. The algorithm proposed here has been applied to a unit-commitment problem with 10^4 variables, see [17].

Finally we mention that our proposal still suffers some weakness concerning speed of convergence: there is a gap between conceptual and implementable forms. For instance, it would be interesting to see whether the results obtained in [8] could be used to fill this gap. On the other hand, our approach does not take advantage of the recent and promising concepts from [19]; for example, we do not use the decomposition of the space introduced in [23]. Yet, such techniques are probably crucial to devising fast algorithms for nonsmooth optimization, as well as defining classes of problems that are amenable to fast resolution. This issue is currently under investigation.

Acknowledgements

We thank the referees for their careful reading and helpful comments.

References

[1] A. Auslender, Numerical methods for nondifferentiable convex optimization, *Mathematical Programming Study* 30 (1987) 102–126.

[2] J. Bonnans, J. Gilbert, C. Lemaréchal and C. Sagastizábal, A family of variable metric proximal methods, *Mathematical Programming* 68 (1995) 15–48.

[3] U. Brännlund, On relaxation methods for nonsmooth convex optimization, Ph.D. thesis, Royal Institute of Technology (Stockholm, 1993).

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

[5] R. Correa and C. Lemaréchal, Convergence of some algorithms for convex minimization, *Mathematical Programming* 62 (1993) 261–275.

[6] J. Dennis and J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review* 19 (1977) 46–89.

[7] M. Fukushima, A descent algorithm for nonsmooth convex optimization, *Mathematical Programming* 30 (1984) 163–175.

[8] M. Fukushima and L. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM Journal on Optimization* 6 (1996) 1106–1120.

[9] O. Güler, On the convergence of the proximal point algorithm for convex minimization, *SIAM Journal on Control and Optimization* 29 (1991) 403–419.

[10] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* (Springer, Berlin, 1993) (two volumes).

[11] J.E. Kelley, The cutting plane method for solving convex programs, *Journal of the Society for Industrial and Applied Mathematics* 8 (1960) 703–712.

[12] K. Kiwiel, Proximity control in bundle methods for convex nondifferentiable minimization, *Mathematical Programming* 46 (1990) 105–122.
[13] K. Kiwiel, Finding normal solutions in piecewise linear programming, *Applied Mathematics and Optimization* (1996) to appear.
[14] C. Lemaréchal, A view of line-searches, in: A. Auslender, W. Oettli and J. Stoer, eds., *Optimization and Optimal Control*, Lecture Notes in Control and Information Science, Vol. 30 (Springer, Heidelberg, 1981) 59–78.
[15] C. Lemaréchal and R. Mifflin, A set of nonsmooth optimization test problems, in: C. Lemaréchal and R. Mifflin, eds., *Nonsmooth Optimization*, (Pergamon Press, Oxford, 1978) 151–165.
[16] C. Lemaréchal, A. Nemirovskii and Y. Nesterov, New variants of bundle methods, *Mathematical Programming* 69 (1995) 111–148.
[17] C. Lemaréchal, F. Pellegrino, A. Renaud and C. Sagastizábal, Bundle methods applied to the unit-commitment problem, in: J. Doleãl and J. Fidler (eds.), *System Modelling and Optimization* (Chapmmann and Hall, 1996) 395–402.
[18] C. Lemaréchal and C. Sagastizábal, An approach to variable metric bundle methods, in: J. Henry and J.-P. Yvon, eds., *Systems Modelling and Optimization*, Lecture Notes in Control and Information Sciences, Vol. 197 (Springer, Berlin, 1993) 144–162 [Also as Rapport de Recherche INRIA #2128 (1994)].
[19] C. Lemaréchal and C. Sagastizábal, Practical aspects of the Moreau–Yosida regularization: theoretical premilinaries, *SIAM Journal on Optimization* 7(4) (1997).
[20] C. Lemaréchal, J.-J. Strodiot and A. Bihain, On a bundle method for nonsmooth optimization, in: O. Mangasarian, R. Meyer and S. Robinson, eds., *Nonlinear Programming*, Vol. 4 (Academic, New York, 1981) 245–282.
[21] B. Martinet, Régularisation d'inéquations variationnelles par approximations successives, *Revue Française d'Informatique et Recherche Opérationnelle* R-3 (1970) 154–179.
[22] R. Mifflin, Semi-smooth and semi-convex functions in constrained optimization, *SIAM Journal on Control and Optimization* 15 (1977) 959–972.
[23] R. Mifflin, A quasi-second-order proximal bundle algorithm, *Mathematical Programming* 73 (1996) 51–72.
[24] J. Moré, Recent developments in algorithms and software for trust region methods, in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming, the State of the Art* (Springer, Berlin, 1983) 258–287.
[25] J. Moreau, Proximité et dualité dans un espace Hilbertien, *Bulletin de la Société Mathématique de France* 93 (1965) 273–299.
[26] M. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: R. Cottle and C. Lemke, eds., *Nonlinear Programming*, SIAM–AMS Proceedings, Vol. 9 (American Mathematical Society, Providence, RI, 1976).
[27] G. Pritchard, G. Gürkan and A. Özge, A note on locally Lipschitzian functions, *Mathematical Programming* 71 (1995) 369–370.
[28] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research* 18 (1993) 227–244.
[29] R. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization* 14 (1976) 877–898.
[30] H. Schramm and J. Zowe, A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results, *SIAM Journal on Optimization* 2 (1992) 121–152.
[31] K. Yosida, *Functional Analysis* (Springer, Berlin, 1964).