IDENTIFYING STRUCTURE OF NONSMOOTH CONVEX FUNCTIONS BY THE BUNDLE TECHNIQUE*

Aris Daniilidis†, Claudia Sagastizábal‡, and Mikhail Solodov§

10th July 2008

ABSTRACT

We consider the problem of minimizing nonsmooth convex functions, defined piecewise by a finite number of functions each of which is either convex quadratic or twice continuously differentiable with positive definite Hessian on the set of interest. This is a particular case of functions with primal-dual gradient structure, a notion closely related to the so-called $\mathcal{VU}$ space decomposition: at a given point, nonsmoothness is locally restricted to the directions of the subspace $\mathcal{V}$, while along the subspace $\mathcal{U}$ the behaviour of the function is twice differentiable. Constructive identification of the two subspaces is important, because it opens the way to devising fast algorithms for nonsmooth optimization (by following iteratively the manifold of smoothness, on which superlinear $\mathcal{U}$-Newton steps can be computed). In this work we show that for the class of functions in consideration, the information needed for this identification can be obtained from the output of a standard bundle method for computing proximal points, provided a minimizer satisfies the nondegeneracy and strong transversality conditions.

Key words. nonsmooth optimization, convex minimization, $\mathcal{VU}$-decomposition, primal-dual gradient structure, partial smoothness, bundle method, proximal point.

AMS subject classifications. 90C25, 65K05, 49J52.

* The first author is supported by the MEC grant MTM2005-08572-C03-03 (Spain). The second author is partially supported by CNPq Grant No. 303540-03/6, by PRONEX-Optimization and by FAPERJ. The third author is supported in part by CNPq Grants 301508/2005-4 and 471267/2007-4, by PRONEX-Optimization and by FAPERJ.

† Departament de Matemàtiques, Universitat Autònoma de Barcelona, E-08193, Bellaterra, Spain.
Email: `arisd@mat.uab.es`
`http://mat.uab.es/~arisd`

‡ CEPEL, Electric Energy Research Center. On leave from INRIA Rocquencourt, France.
Postal address: IMPA, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro RJ 22460-320, Brazil.
Email: `sagastiz@impa.br`
`http://w3.impa.br/~sagastiz/`

§ IMPA – Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil.
Email: `solodov@impa.br`
`http://w3.impa.br/~optim/solodov.html`

1 Introduction

Consider the problem

$$\min_{x \in \Re^n} f(x), \tag{1.1}$$

where the objective function $f : \Re^n \to \Re$ is convex. If f is not differentiable at a solution $\bar{x}$ of (1.1), constructing fast practical algorithms to compute $\bar{x}$ is a challenge. Essentially, this has to do with the intrinsic difficulty in using (or even defining!) appropriate "second-order" objects that capture the behaviour of f around $\bar{x}$. One line of research (e.g., [13, 23, 17, 14, 2]) suggests introducing second-order information about f by means of its Moreau-Yosida regularization, which is a smooth function. The other line of research parts from the viewpoint that nonsmoothness in practical applications is usually "structured" [25, 18, 15, 7, 24]. Nonsmooth functions may behave smoothly and even have appropriate second-order representations on certain manifolds along certain directions. If this structure can be constructively identified and if the relevant manifold can be "followed" iteratively, this opens the potential for designing algorithms with fast local convergence.

In this work, we contribute to the second line of research, and in particular to constructing the so-called $\mathcal{VU}$-decomposition [12] for functions with primal-dual gradient (PDG) structure [18, 19, 20, 21] (see Section 2 below for definitions and a summary of relevant details). This development is important for the following reasons. In [19, 21] it is shown that if f has PDG structure and f satisfies at $\bar{x}$ the strong transversality condition stated in (2.2) below, then for points close to $\bar{x}$ the proximal map generates points on the manifold of smoothness $\mathcal{M}$, called "fast track". Since proximal points can be approximated arbitrarily well by bundle techniques, [22] proposes a fast $\mathcal{VU}$-algorithm that performs a corrector-predictor step at each iteration. More precisely, by means of the bundle subroutine, a (corrector) proximal step is made in order to bring the iterate to the fast track $\mathcal{M}$. Then the $\mathcal{U}$-Newton (predictor step) is performed to gain superlinear decrease of the distance to solution. A geometrical study of such methods, including relations with Sequential Quadratic Programming, can be found in [16]. While computing the $\mathcal{U}$-Newton step certainly requires approximating the proximal point well enough, [22] shows that this computational effort can be worthwhile when compared to standard bundle methods that stop this approximation much earlier (as soon as sufficient descent of the objective function is attained). This is especially so in cases where high precision is required. Standard forms of bundle methods may be quite slow (even sublinear) when approaching a solution, which makes obtaining high precision impossible. This is where the $\mathcal{U}$-Newton superlinear steps are most important. We refer the reader to [22] for a comparison of an overall computational behaviour of a usual bundle method and a $\mathcal{VU}$-algorithm, where practical superlinear convergence of the latter had been verified.

While [22] suggests a way of generating a basis for $\mathcal{U}$ in the process of computing the proximal point by the bundle subroutine, the fact that this construction is "correct" is essentially stated as an assumption, albeit a clearly reasonable one. In what follows, we prove that for the given class of functions, if x is close enough to $\bar{x}$ then the subspace $\mathcal{V}$ (and, hence, also $\mathcal{U}$) at the proximal point p of x can indeed be recovered from the objects generated by the bundle subroutine in the process of computing p. Numerical results presented in Section 4 confirm this assertion.

Our notation is fairly standard. We shall denote by $B_\varepsilon(\bar{x})$ the open ball with center $\bar{x} \in \Re^n$ and radius $\varepsilon > 0$. For $x, y \in \Re^n$, $\langle x, y \rangle$ stands for the inner product of x and y. Given a convex function $f : \Re^n \to \Re$, we denote by $\partial f(x)$ its subdifferential at the point $x \in \Re$:

$$\partial f(x) = \{g \in \Re^n : f(y) - f(x) \geq \langle g, y - x \rangle, \quad \text{for all } y \in \Re^n\}. \tag{1.2}$$

The canonical simplex in the space $\Re^s$ will be denoted by

$$\Delta_s = \left\{ t \in \Re^s : t \geq 0\,, \sum_{i=1}^{s} t_i = 1 \right\}.$$

For a convex set C, by $\operatorname{ri} C$ we denote its relative interior. For any set $C \subset \Re^n$, $\operatorname{lin} C$ stands for its linear hull (the smallest subspace of $\Re^n$ that contains C) and $\operatorname{aff} C$ for its affine hull (the smallest affine manifold of $\Re^n$ that contains C). Finally, the cardinality of a (finite) set I is denoted by $|I|$.

2 Analytic description of the function structure

We consider the class of convex functions defined piecewise by a finite collection of twice continuously differentiable convex functions. Specifically, for all $x \in \Re^n$,

$$\begin{gathered} f(x) \in \{f_j(x),\ j = 0, \ldots, m\}, \\ \text{where } f \text{ is convex and } f_j : \Re^n \to \Re,\ j = 0, \ldots, m, \text{ are convex of class } C^2. \end{gathered} \tag{2.1}$$

We shall refer to the functions f_j, $j = 0, 1, \ldots, m$, as *structure functions*. For some of our results we shall eventually assume that, in addition to the above, each structure function f_j is either quadratic or has positive definite Hessian in a relevant neighbourhood of a minimizer of f. Those assumptions do not introduce any restrictions that are truly relevant with respect to the task at hand – identification of the smoothness structure of (structured) nonsmooth functions.

A classical example of (2.1) is the max-function $f(x) = \max_{j=0,\ldots,m} f_j(x)$, where f_j are convex of class C^2. However, the class given by (2.1) is not restricted to max-functions.

2.1 Primal-dual gradient structure and space decomposition

Given a convex function of the form (2.1), its subdifferential at a point $x \in \Re^n$ can be computed in terms of the derivatives of the structure functions that are active at x. More precisely,

$$\partial f(x) = \Big\{ g \in \Re^n : g = \sum_{j \in I(x)} t_j f_j'(x),\ t \in \Delta_{|I(x)|} \Big\},$$

where

$$I(x) = \Big\{ j \in \{0, \ldots, m\} : f(x) = f_j(x) \Big\}$$

is the set of "active" indices at x. Let $\bar{x} \in \Re^n$ be a solution to (1.1), where f has the form of (2.1). By continuity of the structure functions, there exists a ball $B_\varepsilon(\bar{x}) \subseteq \Re^n$ such that

$$\forall\, x \in B_\varepsilon(\bar{x}), \quad I(x) \subseteq I(\bar{x}).$$

For convenience, we assume that the cardinality of $I(\bar{x})$ is m_1+1 and reorder the structure functions, if necessary, so that $I(\bar{x}) = \{0, 1, \dots, m_1\}$. From now on, we consider that

$$\forall\, x \in B_\varepsilon(\bar{x}), \quad f(x) \in \{f_j(x),\ j = 0, \dots, m_1\}.$$

The class of functions (2.1) belongs to the PDG-structured family ([21]; see also [18, 20]). More precisely, following the terminology and notation of [21], a function f satisfying (2.1) has a PDG structure at $\bar{x}$ relative to the set $B_\varepsilon(\bar{x})$ with primal functions $f_j,\ j = 0, \dots, m_1$, and dual multiplier set Δ_{m_1+1}. We note that PDG structures are closely related to the so-called $\mathcal{VU}$-space decomposition (see [12, 18, 20, 21]), which shall be the focus of our analysis. Given a point $x \in \Re^n$ and any subgradient $g \in \partial f(x)$, the $\mathcal{VU}$-space decomposition at x is given by

$$\mathcal{V}(x) = \operatorname{lin}\{\partial f(x) - g\}, \quad \mathcal{U}(x) = \mathcal{V}(x)^\perp.$$

The nonsmoothness of the function f at x is reflected by its V-shaped graph along the subspace $\mathcal{V}$, while along the subspace $\mathcal{U}$ the function appears to behave smoothly [12]. Roughly speaking, the function f is “partly smooth” with respect to some “active” manifold $\mathcal{M}$ containing $\bar{x}$, in a way that for every $x \in \mathcal{M}$ the $\mathcal{U}$-space of the $\mathcal{VU}$-space decomposition at x is the tangent space of the manifold $\mathcal{M}$ at x (see the precise terminology in [15], and also [16, 4] for more details).

When the function f has a PDG structure (thus, in particular, in the case of (2.1)), for every $x \in \Re^n$ and any fixed $l \in I(x)$ it holds that

$$\mathcal{V}(x) = \operatorname{lin}\{f_j'(x) - f_l'(x),\ j \in I(x)\}.$$

We say that f satisfies at $\bar{x}$ the condition of *strong transversality* if

$$\text{the set } \{\, f_j'(\bar{x}) - f_0'(\bar{x}),\ j = 1, \dots, m_1\} \text{ is linearly independent.} \tag{2.2}$$

The following properties are consequences of strong transversality:

- The set
$$\{f_j'(\bar{x}) - f_0'(\bar{x}),\ j = 1, \dots, m_1\}$$
is a basis for the subspace $\mathcal{V}(\bar{x})$ (of dimension $\dim \mathcal{V}(\bar{x}) = |I(\bar{x})| - 1 = m_1$).

- For any $x \in B(\bar{x})$ and any fixed $l \in I(x)$, the set
$$\{f_j'(x) - f_l'(x),\ j \in I(x) \setminus \{l\}\}$$
is linearly independent and forms a basis for the subspace $\mathcal{V}(x)$ (of dimension $\dim \mathcal{V}(x) = |I(x)| - 1 \le m_1$).

- For all $x \in B(\bar{x})$, “interior” subgradients are generated by “interior simplicial multipliers”, in the sense that for any $p \in B_\varepsilon(\bar{x})$ such that $I(p) = I(\bar{x})$, it holds that
$$\operatorname{ri} \partial f(p) := \Big\{g \in \Re^n : g = \sum_{j=0}^{m_1} t_j f_j'(p),\ t \in \Delta_{m_1+1}\,, t > 0\Big\}, \tag{2.3}$$
see [10, Remark III.2.1.4].

The main motivation of this work is to provide some building blocks in order to design (and implement) fast $\mathcal{VU}$-algorithms. Our objective is to determine a basis for the $\mathcal{V}$-space (thus also $\mathcal{U}$-space) at points p satisfying $I(p) = I(\bar{x})$, so that a superlinear $\mathcal{U}$-Newton step can be computed. Under the strong transversality condition (2.2), locally the set of such points coincides with the active manifold $\mathcal{M}$ which, under the nondegeneracy condition (see (2.5) below), contains proximal points of points that are close enough to the solution $\bar{x}$. As a consequence, this manifold can be iteratively followed by an implementable algorithm and be combined with superlinear $\mathcal{U}$-Newton steps, as will be explained in the sequel.

2.2 Connections with smooth manifolds and proximal points

Since the subspaces $\mathcal{U}$ and $\mathcal{V}$ generate the whole space $\Re^n$, every vector can be decomposed along its $\mathcal{VU}$-components. In particular, any $z \in \Re^n$ can be expressed as

$$\Re^n \ni z \quad = \quad z_{\mathcal{V}(\bar{x})} \oplus z_{\mathcal{U}(\bar{x})} \in \Re^{dim\,\mathcal{V}(\bar{x})} \times \Re^{dim\,\mathcal{U}(\bar{x})}.$$

As is shown in [21, Thm. 3.1], a PDG-structured function that satisfies the strong transversality condition at $\bar{x}$, has a smooth primal track which assigns for each sufficiently small $u \in \mathcal{U}(\bar{x})$ an element $\chi(u)$ of the active manifold $\mathcal{M}$. Since along the track the $\mathcal{U}(\chi(u))$-component of the subdifferential of f is a singleton [21, Thm. 4.1(*ii*)], the restriction of the function f along the track (active manifold) appears to be smooth. (Moreover, let us recall that in [19] the smooth primal track from [21] allowing a second order expansion of f along the $\mathcal{U}$-subspace was called “fast track”.)

The active manifold $\mathcal{M}$ is theoretically defined as follows:

$$\mathcal{M} = \{p \in B_\varepsilon(\bar{x}) : f_j(p) = f(p),\, j = 0, 1, \ldots, m_1\}\,, \tag{2.4}$$

or equivalently,

$$\mathcal{M} = \{p \in B_\varepsilon(\bar{x}) \ : I(p) = I(\bar{x})\}\,.$$

Note that the latter, in view of (2.2), yields that $\mathcal{M}$ is a smooth manifold. It is easy to verify that in this case f is also partly smooth with respect to the manifold $\mathcal{M}$, according to the terminology introduced in [15] (see [16] for details). Furthermore, the Riemanian gradient of the restriction of f on $\mathcal{M}$ at a point $x = \chi(u) \in \mathcal{M}$ is the projection of the subdifferential $\partial f(x)$ on the $\mathcal{U}$-space at x, while the normal cone $N_{\mathcal{M}}(x)$ is the $\mathcal{V}(x)$-subspace.

For all p close enough to $\bar{x}$ it holds ([7, Thm. 5.15] and [21, Thm. 5.3]) that

$$p \in \mathcal{M} \iff p = \chi(u(p)) \text{ with } u(p) = (p - \bar{x})_{\mathcal{U}(\bar{x})}\,.$$

It follows that the concepts “fast track” $\chi(u)$ and “smooth manifold” $\mathcal{M}$ describe in a different manner the same object, that identifies the smoothness structure of f. However, since both $\chi(u)$ and $\mathcal{M}$ are defined implicitly, their computation is far from being straightforward. It is at this stage that the important connection with proximal points comes into play, as discussed next.

In the sequel, the following notion will be needed. We say that f satisfies at its minimizer $\bar{x}$ the *nondegeneracy* condition if

$$0 \in \operatorname{ri} \partial f(\bar{x}). \tag{2.5}$$

This nondegeneracy condition is not very restrictive. The situation when a given minimizer of f has 0 in the "extreme boundary" of its subdifferential, as for instance for the function

$$f(x) = \begin{cases} x_1 & \text{if } x_1 \geq 0, \\ 0 & \text{if } x_1 < 0, \end{cases} \qquad x \in \Re^n,$$

where $\partial f(0) = [0,1] \times \{0_{n-1}\}$, is unstable and generically not present in practice. On the other hand, condition (2.5) ensures a certain stability in the behaviour of the subdifferential mapping. Indeed, it has been shown that if f is a strongly transversal PDG-structured lower semicontinuous (respectively, convex) function, the condition stated in (2.5) is transmitted to some specific continuous selections of subgradients (parameterized by u), see [21, Thm. 4.2] (respectively, [19, Thm. 5.2]). More specifically, the following holds (see [4, Lemma 20]).

Lemma 2.1.1. (Persistence and stability of the nondegeneracy condition) *Let f be a convex PDG-structured function, and let $\mathcal{M}$ be the active manifold along which f admits a fast track.*

Then for any continuous selection

$$p \mapsto g(p), \quad p \in \mathcal{M}$$

of the affine space mapping

$$p \mapsto \operatorname{aff}(\partial f(p)), \quad p \in \mathcal{M}$$

that satisfies $g(\bar{x}) \in \operatorname{ri} \partial f(\bar{x})$ for $\bar{x} \in \mathcal{M}$, it holds that $g(p) \in \operatorname{ri} \partial f(p)$ for all $p \in \mathcal{M}$ near $\bar{x}$. □

Given a point $x \in \Re^n$ and a *prox-parameter* $\mu > 0$, the proximal point of f at x, denoted by $p_\mu(x)$, is given by

$$p_\mu(x) = \arg \min_{y \in \Re^n} \left\{ f(y) + \frac{\mu}{2} \|y - x\|^2 \right\}.$$

(Clearly, the mapping p_μ is single-valued, by the strict convexity of the norm.)

The relevance of proximal points in our context is two-fold.

- When x is sufficiently close to $\bar{x}$ and the nondegeneracy condition (2.5) holds, $p_\mu(x)$ lies on the fast track (active manifold $\mathcal{M}$); see [19, 7, 21, 4]. Since $p_\mu(x) \in \mathcal{M}$, by definition of the smooth manifold, we see that

$$I(p_\mu(x)) = I(\bar{x}) \quad \text{or, equivalently,} \quad f(p_\mu(x)) = f_j(p_\mu(x)),\, j = 0, 1, \ldots, m_1\,. \tag{2.6}$$

- A sequence of null steps of a bundle subroutine can approximate proximal points within any desired accuracy [3] (see also [9] for the nonconvex case).

Therefore, under the condition (2.5), the theoretical concepts of *fast track* and *active manifold* of a partly smooth function f can be locally related to the set of proximal points of f, opening the way to implementable and fast $\mathcal{VU}$-algorithms. Essentially, these algorithms follow a trajectory of proximal points leading to $\bar{x}$, with superlinear Newtonian acceleration at those points. This acceleration is possible due to the fact that proximal points belong to the active manifold $\mathcal{M}$ on which f has second-order expansion along $\mathcal{U}(p)$, and so at such points a fast (superlinear) $\mathcal{U}$-Newton step can be computed, provided the subspaces $\mathcal{V}$ and $\mathcal{U}$ are known, or sufficiently well approximated [22].

3 Identifying structure by the bundle technique

We proceed to analyze the computation of the proximal point $p_\mu(x)$ of a given $x \in B_\varepsilon(\bar{x})$ for a convex function f of the form (2.1). We show that a basis for the subspace $\mathcal{V}$ at $p_\mu(x)$ is obtained as a by-product of computing $p_\mu(x)$ by the bundle technique.

We start with the following remark: complete knowledge of the subdifferential $\partial f(p)$ at the proximal point $p = p_\mu(x) \in \mathcal{M}$ is certainly sufficient for determining the $\mathcal{V}$-space of $\mathcal{VU}$-decomposition at p. However, such information is considered prohibited (impossible to obtain) in practice. Indeed, apart from the point p being unknown (it needs to be computed by an iterative procedure), the typical practical requirement in computational nonsmooth optimization (referred to as *black-box* information, see, e.g., [1, Part II]) gives access to only one subgradient at each point, and not to the whole subdifferential. Information about the relevant subspaces of the $\mathcal{VU}$-decomposition should therefore be built iteratively, in the process of computing the proximal point. The practical way of computing proximal points is the bundle method [11, 10, 1]. We next show how this procedure can be used to build a basis for the $\mathcal{V}$-space at p. Let us first state formally what we mean by the black-box information, specifically for a convex function f of the form (2.1):

$$\text{Given } x^i \in \Re^n \text{ (input), an arbitrary index } j_i \text{ in } I(x^i) \text{ is available (in principle, only one).} \tag{3.1}$$

This information gives one affine function

$$f_{j_i}(x^i) + \langle f'_{j_i}(x^i), y - x^i \rangle, \quad y \in \Re^n,$$

which supports the graph of f from below. The bundle method approximates the proximal point p of x by iteratively computing proximal points of the cutting-plane approximations of f defined by the previously accumulated affine functions. Specifically, if $x^0 (= x), x^1, \ldots, x^{k-1}$ are the previous iterates, then the k-th iterate is given by

$$x^k = \arg\min_{y \in \Re^n} \left\{ \psi_{k-1}(y) + \frac{\mu}{2}\|y - x\|^2 \right\}, \tag{3.2}$$

where

$$\psi_{k-1}(y) \ = \max_{i=0,1,\ldots,k-1} \Big\{ \ f_{j_i}(x^i) + \langle f'_{j_i}(x^i), y - x^i \rangle \ , \quad j_i \in I(x^i) \ \Big\}, \tag{3.3}$$

is the cutting-planes model of f. This problem is solved via its quadratic programming (QP) reformulation

$$\begin{array}{ll} \min_{(y,r)\in\Re^{n+1}} & \{ r + \frac{\mu}{2}\|y - x\|^2 \} \\ \text{s.t.} & f_{j_i}(x^i) + \langle f'_{j_i}(x^i), y - x^i \rangle \leq r, \ i = 0, 1, \ldots, k-1. \end{array}$$

By the optimality condition for (3.2), it holds that

$$x^k = x - \frac{1}{\mu} g^k, \quad \text{where } g^k \in \partial \psi_{k-1}(x^k). \tag{3.4}$$

Since the model ψ_{k-1} is a convex max-function, its subgradients at x^k are convex combinations of the derivatives of its active pieces at x^k, with coefficients given by the multipliers (dual solutions) of the QP. Eliminating all the indices corresponding to zero multipliers (including active ones, if there exist zero multipliers corresponding to active QP constraints), we can write

$$g^k = \sum_{i \in \tilde{I}_k} t_i^k f'_{j_i}(x^i), \quad t^k \in \Delta_{|\tilde{I}_k|}, \quad t^k > 0, \tag{3.5}$$

where

$$\begin{array}{rcl} \tilde{I}_k & = & \Big\{ i \in \{0, 1, \ldots, k-1\} : \; t_i^k > 0 \Big\} \\ & \subset & \Big\{ i \in \{0, 1, \ldots, k-1\} : \; \psi_{k-1}(x^k) = f_{j_i}(x^i) + \langle f'_{j_i}(x^i), x^k - x^i \rangle \Big\}. \end{array}$$

We note that to ensure convergence, for the $(k+1)$-iteration it is sufficient to keep in the bundle memory only those affine functions that correspond to indices in $\tilde{I}_k$ at the k-th iteration, permanently deleting all the rest (but adding the new affine function computed at x^k). Without introducing this feature explicitly in the analysis, we shall make the following (related) assumption:

$(\mathcal{H})$ The cardinality of $\tilde{I}_k$, $k = 0, 1, \ldots$, is uniformly bounded in k.

While there is no formal argument to justify this assumption, as a practical matter it is very natural. In fact, many (active set) QP solvers choose linearly independent bases, i.e., work with "minimal" representations. In the representation of $g^k \in \Re^n$ in (3.5), this means that QP solver gives a solution such that $|\tilde{I}_k| \leq n + 1$ (such a solution always exists by the Carathéodory Theorem). A similar assumption/property for a QP solver had been used, for a different QP-based method, in [6, Sec. 5].

Our development below relies on the PDG structure of f at $\bar{x}$ relative to $B_\varepsilon(\bar{x})$. For this reason, we first ensure that iterates do not leave the relevant set.

Proposition 3.1.1. (Localization of the bundle iterates)
Let f be a convex function and let $\{x^k\}$ be a sequence generated according to (3.2)–(3.3), with $x^0 = x \in B_\varepsilon(\bar{x})$.

Then for every $\delta \in (0, \varepsilon)$ there exists $\bar{\mu} > 0$ such that if $\mu \geq \bar{\mu}$ and $x \in B_{\varepsilon-\delta}(\bar{x})$ then $\{x^k\} \subset B_\varepsilon(\bar{x})$.

Furthermore, if f satisfies at $\bar{x}$ the nondegeneracy condition (2.5) then

$$I(x^k) \subseteq I(\bar{x}) = I(p_\mu(x)) \quad \textit{for all } k. \tag{3.6}$$

Proof. For any index k, the cutting-plane model ψ_k given in (3.3) has the same Lipschitz constant as f, say $L > 0$. Since, by (3.4), $g^k = \mu(x - x^k) \in \partial\psi_{k-1}(x^k)$, this means that $\mu\|x^k - x\| \leq L$ for all k. Hence,

$$\|x^k - \bar{x}\| \leq \|x^k - x\| + \|x - \bar{x}\| \leq \frac{L}{\mu} + \varepsilon - \delta,$$

and the assertion follows taking $\mu \geq \bar{\mu} = L/\delta$.

The inclusion in (3.6) follows from the continuity of the structure functions, while the equality is a consequence of (2.5) (see (2.6)). ■

From now on, the prox-parameter $\mu > 0$ is assumed to be sufficiently large to ensure that $\{x^k\} \subset B_\varepsilon(\bar{x})$ and (3.6) holds. Therefore, any bundle index $j_i \in I(x^i)$ belongs to the set $I(\bar{x}) = \{0, 1, \ldots, m_1\}$. For each k, we define the "accumulation" of the simplicial multipliers in (3.5), corresponding to the same structure function f_l as follows:

$$q_l^k = \sum_{i \in \tilde{I}_k \,;\, j_i = l} t_i^k, \qquad l = 0, 1, \ldots, m_1. \tag{3.7}$$

(We formally set the result of summing up over an empty set to be 0.) Clearly, these multipliers satisfy

$$q^k := (q_0^k, \ldots, q_{m_1}^k) \in \Delta_{m_1+1} \quad \text{for all } k.$$

The following result concerns asymptotic approximation of the specific subgradient $\mu(x-p) \in \partial f(p)$, $p = p_\mu(x)$, by the sequence $\{g^k\}$ produced by the bundle procedure.

Proposition 3.1.2. (Asymptotic behaviour of the bundle procedure) *Let f be a convex function of the form* (2.1) *and assume that it satisfies at $\bar{x}$ the nondegeneracy condition* (2.5).

Let a sequence $\{x^k\}$ be generated according to (3.2)–(3.3)*, and let $p = p_\mu(x)$. Then for all $k \geq 0$ there exist $\tau_i \in [0,1]$, $i \in \tilde{I}_k$, such that*

$$\sum_{i \in \tilde{I}_k} t_i^k \langle f''_{j_i}(x^i + \tau_i(p - x^i))(p - x^i), p - x^i \rangle \to 0 \text{ as } k \to \infty, \tag{3.8}$$

$$g^k = \sum_{l=0}^{m_1} q_l^k f'_l(p) - \sum_{i \in \tilde{I}_k} t_i^k \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i) d\theta \to \mu(x - p) \text{ as } k \to \infty. \tag{3.9}$$

Proof. As is well known (e.g., [3, Prop. 4.1]), for iterates generated by the bundle procedure it holds that $\psi_{k-1}(x^k) \nearrow f(p)$ and $x^k \to p$ as $k \to \infty$. Therefore, taking into account also that the sequence $\{g^k\}$ is evidently bounded (by (3.4) and Proposition 3.1.1), we deduce that

$$\lim_{k \to \infty} \Big(\psi_{k-1}(x^k) + \langle g^k, p - x^k \rangle \Big) = f(p). \tag{3.10}$$

By the definition of $\tilde{I}_k$, and since $t^k \in \Delta_{|\tilde{I}_k|}$, we have that

$$\psi_{k-1}(x^k) = \sum_{i \in \tilde{I}_k} t_i^k (f_{j_i}(x^i) + \langle f'_{j_i}(x^i), x^k - x^i \rangle).$$

Together with (3.5), this gives

$$\begin{aligned} \psi_{k-1}(x^k) + \langle g^k, p - x^k \rangle &= \sum_{i \in \tilde{I}_k} t_i^k (f_{j_i}(x^i) + \langle f'_{j_i}(x^i), x^k - x^i + p - x^k \rangle) \\ &= \sum_{i \in \tilde{I}_k} t_i^k (f_{j_i}(x^i) + \langle f'_{j_i}(x^i), p - x^i \rangle). \end{aligned} \tag{3.11}$$

By the Mean-Value Theorem, for each $i \in \tilde{I}_k$ there exists $\tau_i \in [0,1]$ such that

$$\begin{array}{rcl} f_{j_i}(x^i) + \langle f'_{j_i}(x^i), p - x^i \rangle & = & f_{j_i}(p) - \frac{1}{2}\langle f''_{j_i}(x^i + \tau_i(p - x^i))(p - x^i), p - x^i \rangle \\ & = & f(p) - \frac{1}{2}\langle f''_{j_i}(x^i + \tau_i(p - x^i))(p - x^i), p - x^i \rangle, \end{array}$$

where we have used that $f_{j_i}(p) = f(p)$ (By (2.6) and the fact that, by Proposition 3.1.1, we have that $j_i \in I(p)$ for all $i \in \tilde{I}_k$ and all $k \geq 0$).

Combining now the latter relation with (3.11), we obtain that

$$\begin{array}{rcl} \psi_{k-1}(x^k) + \langle g^k, p - x^k \rangle & = & \sum_{i \in \tilde{I}_k} t_i^k (f(p) - \frac{1}{2}\langle f''_{j_i}(x^i + \tau_i(p - x^i))(p - x^i), p - x^i \rangle) \\ & = & f(p) - \frac{1}{2} \sum_{i \in \tilde{I}_k} t_i^k \langle f''_{j_i}(x^i + \tau_i(p - x^i))(p - x^i), p - x^i \rangle, \end{array}$$

where we have used again that $t^k \in \Delta_{|\tilde{I}_k|}$. Relation (3.8) now follows from (3.10).

By the Mean-Value Theorem (for vector functions), for each $i \in \tilde{I}_k$ we can also write

$$f'_{j_i}(p) = f'_{j_i}(x^i) + \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i) d\theta.$$

Then, using (3.5), we have that

$$\begin{array}{rcl} g^k & = & \sum_{i \in \tilde{I}_k} t_i^k \left(f'_{j_i}(p) - \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i) d\theta \right) \\ & = & \sum_{l=0}^{m_1} q_l^k f'_l(p) - \sum_{i \in \tilde{I}_k} t_i^k \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i) d\theta. \end{array}$$

Since $x^k \to p$ as $k \to \infty$, from (3.4) we have that $g^k = \mu(x - x^k) \to \mu(x - p)$, and (3.9) follows. ■

Until now our analysis did not require any assumptions on the structure functions f_j other than twice continuous differentiability. For our main result, concerning the construction of a basis for the subspace $\mathcal{V}$ by means of the active bundle gradients, we assume that each structure function is either quadratic or its Hessian is positive definite (on the set of interest).

Theorem 3.2. (Asymptotic determination of the $\mathcal{V}$-space)
Let f be a convex function of the form (2.1)*, and suppose that it satisfies at $\bar{x}$ the nondegeneracy condition* (2.5) *and the strong transversality condition* (2.2)*. Suppose further that each structure function f_j, $j = 0, 1, \ldots, m_1$, is either quadratic or its Hessian is positive definite on the relevant set $B_\varepsilon(\bar{x})$. Assume, finally, that the hypothesis $(\mathcal{H})$ is satisfied.*

Then the representation (3.5) *for g^k (output of the bundle procedure) provides asymptotically a particular basis of the $\mathcal{V}$-space at $p = p_\mu(x)$ (thus, implicitly, also of the $\mathcal{U}$-space at p), in the sense that*

$$\mathcal{V}(p) = \mathrm{lin}\{v_l/s_l - v_0/s_0,\ l = 1, \ldots, m_1\},$$

where

$$v_l = \lim_{k\to\infty} \sum_{i\in\tilde{I}_k;\, j_i=l} t_i^k f'_{j_i}(x^i), \quad 0 < s_l = \lim_{k\to\infty} \sum_{i\in\tilde{I}_k;\, j_i=l} t_i^k, \qquad l = 0, 1, \dots, m_1.$$

Proof. Let $x \in \Re^n$ be sufficiently close to $\bar{x}$ and $\mu > 0$ be sufficiently large, so that the assertions of Propositions 3.1.1 and 3.1.2 hold.

As already noted,

$$g^k = \sum_{i\in\tilde{I}_k} t_i^k f'_{j_i}(x^i) \to \mu\,(x-p) \,\in\, \mathrm{ri}\,\partial f(p),$$

where the inclusion holds by Lemma 2.1.1 (under the nondegeneracy condition (2.5), for x close enough to $\bar{x}$). Taking now into account (2.3), that holds under the strong transversality condition (2.2), we have that

$$\mu(x-p) = \sum_{j=0}^{m_1} s_j f'_j(p),\ s \in \Delta_{m_1+1},\ s > 0, \tag{3.12}$$

where the "simplicial" multiplier vector $s > 0$ is uniquely defined.

The key idea is to show that the contribution of gradients of all those pieces $i \in \tilde{I}_k$ that are active at the (unknown) proximal point p, is present and "asymptotically positive" in the representation of g^k in (3.9). Specifically, with the notation of (3.7), the sequence $\{q_l^k\}$ converges to a strictly positive number for each $l = 0, 1, \dots, m_1$, while the second term in (3.9) vanishes.

Since, by the convexity of all structure functions f_j, the matrices $f''_j(\cdot)$ are positive semidefinite, we have that all the terms in the sum in (3.8) of Lemma 3.1.2 are nonnegative. Since the sum tends to zero, it then follows that each of those terms tends to zero:

$$\forall\, i \in \tilde{I}_k, \quad 0 = \lim_{k\to\infty} t_i^k \langle f''_{j_i}(x^i + \tau_i(p-x^i))(p-x^i), p-x^i\rangle. \tag{3.13}$$

We next show that

$$\forall\, i \in \tilde{I}_k, \quad 0 = \lim_{k\to\infty} t_i^k \int_0^1 f''_{j_i}(x^i + \theta(p-x^i))(p-x^i)d\theta. \tag{3.14}$$

For $i \in \tilde{I}_k$ such that $\lim_{k\to\infty} t_i^k = 0$, the relation in (3.14) is obvious (since the other term in the product is evidently bounded).

Consider now $i \in \tilde{I}_k$ such that $\liminf_{k\to\infty} t_i^k > 0$. For such i, the relation (3.13) implies that

$$\langle f''_{j_i}(x^i + \tau_i(p-x^i))(p-x^i), p-x^i\rangle = 0. \tag{3.15}$$

For each $j \in \{0, 1, \dots, m_1\}$ such that $j = j_i$, $i \in \tilde{I}_k$, occurs infinitely often, we next consider separately the case when f_j is quadratic and the case when the Hessian of f_j is positive definite.

Case 1 ($f_{j_i}(\cdot)$ is quadratic). Let

$$f_{j_i}(x) = \frac{1}{2}\langle A_{j_i}x, x\rangle + \langle a_{j_i}, x\rangle + c_{j_i},$$

where $c_{j_i} \in \Re$, $a_{j_i} \in \Re^n$ and A_{j_i} is an $n \times n$ symmetric positive semidefinite matrix. Since $f''_{j_i}(\cdot) = A_{j_i}$, we obtain from (3.15) that

$$\langle A_{j_i}(p - x^i), p - x^i\rangle = 0,$$

which means that $p - x^i$ is a minimizer of the nonnegative quadratic form $\langle A_{j_i}y, y\rangle$, $y \in \Re^n$. Hence, its gradient is zero at this point:

$$2A_{j_i}(p - x^i) = 0.$$

Recalling again that $f''_{j_i}(\cdot) = A_{j_i}$, this implies (3.14).

Case 2 ($f''_{j_i}(\cdot)$ is positive definite). Under this assumption, we immediately obtain from (3.15) that

$$p - x^i = 0,$$

which again implies (3.14).

Recalling now that $|\tilde{I}_k|$ is uniformly bounded (assumption ($\mathcal{H}$)), and summing up (3.14) for all $i \in \tilde{I}_k$, we obtain that

$$0 = \lim_{k\to\infty} \sum_{i\in\tilde{I}_k} t_i^k \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i)d\theta,$$

i.e., the second term in (3.9) asymptotically vanishes, and we have that

$$\mu(x - p) = \lim_{k\to\infty} g^k = \lim_{k\to\infty} \sum_{l=0}^{m_1} q_l^k f'_l(p). \tag{3.16}$$

As the multiplier s in (3.12) is unique, it then follows from (3.16) and (3.12) that

$$\lim_{k\to\infty} q_l^k = s_l > 0, \quad l = 0, 1, \ldots, m_1. \tag{3.17}$$

Moreover, using the Mean Value Theorem, (3.17) and (3.14), we have that for all $l = 0, 1, \ldots, m_1$, it holds that

$$\begin{aligned} s_l f'_l(p) &= \lim_{k\to\infty} q_l^k f'_l(p) \\ &= \lim_{k\to\infty} \sum_{i\in\tilde{I}_k;\, j_i=l} t_i^k f'_{j_i}(p) \\ &= \lim_{k\to\infty} \left(\sum_{i\in\tilde{I}_k;\, j_i=l} t_i^k \left(f'_{j_i}(x^i) + \int_0^1 f''_{j_i}(x^i + \theta(p - x^i))(p - x^i)d\theta \right) \right) \\ &= \lim_{k\to\infty} \sum_{i\in\tilde{I}_k;\, j_i=l} t_i^k f'_{j_i}(x^i) = v_l. \end{aligned}$$

Given that $\mathcal{V}(p) = \text{lin}\{f'_l(p) - f'_0(p),\ l = 1, \ldots, m_1\}$, and taking into account that $s_l > 0$ for all $l = 0, 1, \ldots, m_1$, this verifies our claim. ■

4 Numerical benchmark on max-type functions

In order to assess from a practical point of view the theoretical statement of Theorem 3.2, we hereby present some numerical results on a collection of functions having the form (2.1).

We consider 180 randomly generated functions, which are all defined as pointwise maxima of a finite collection of convex quadratic functions. The test-set is defined in such a way that

– for each problem the minimum is attained at $\bar{x} = 0 \in \Re^n$;

– strong transversality (2.2) and nondegeneracy (2.5) are satisfied at $\bar{x}$.

We have performed numerical tests using seven different variants of the stopping rules for approximating the proximal point. The results are reported in a form of tables and performance profiles for different criteria, including accuracy and number of iterations.

4.1 Generating Strongly Transversal Structured Functions

We have used the test-set in the `max-quad` family created in [8][1], with test functions defined as pointwise maximum of a finite collection of quadratic functions:

$$f(x) := \max\left\{ f_j(x) := \frac{1}{2}\langle A_j x, x\rangle + \langle a_j, x\rangle + c_j, \quad j = 0, 1, \ldots, m \right\}, \tag{4.1}$$

where A_j are $(n \times n)$-positive definite matrices, $a_j \in \Re^n$ and $c_j \in \Re$. This family of functions belongs to the class considered in (2.1) and allows to create many different examples by choosing the dimension of the space n, the number of structure functions m, and then randomly generating m objects A_j, a_j and c_j (determining the structure functions). In this setting, taking $\bar{x} = 0 \in \Re^n$ and fixing the dimension m_1 of the $\mathcal{V}(\bar{x})$-space, we have (reordering indices if necessary) that

$$f(0) = c_j = C \text{ for } j = 0, 1, \ldots m_1, \quad c_j < C \text{ for } j = m_1 + 1, \ldots, m,$$

$$\partial f(0) = \operatorname{conv}\{a_j : j = 0, \ldots, m_1\}.$$

If the random vectors a_j, $j = 0, \ldots, m_1$, are generated so that they are affinely independent and $\sum_{j=0}^{m_1} \bar{t}_j a_j = 0$ for some $\bar{t} > 0$, then conditions (2.2) and (2.5) are satisfied.

For comparison purposes, we generate problems for which the proximal point p and the $\mathcal{V}$-space at p are computed *a priori*. We proceed as follows. For each function, we start by fixing the desired proximal point to be a small vector p, close enough to $\bar{x} = 0$, and satisfying (2.6). We then set the proximal parameter $\mu = 1.01\,L + 1$, for the Lipschitz constant

$$L = \max\{\|A_j\| : j = 0, 1, \ldots, m\}.$$

Having this information, we take as starting point of the iterative process

$$x = x^0 = p + \frac{1}{\mu}\gamma \quad \text{for } \gamma \in \operatorname{conv}\{A_j p + a_j : j = 0, \ldots, m_1\}, \tag{4.2}$$

[1] The authors wish to thank Warren Hare for providing the starting MATLAB blocks for the `max-quad` family

a choice equivalent to having $p = p_\mu(x^0)$. We further set the values of ε and δ (*cf.* Proposition 3.1.1) equal to $\|x^0\|$ and $\|x^0\|/2$, respectively, and we check whether $\mu \geq 2L/\|x^0\|$ (which guarantees that Proposition 3.1.1 holds true). If this is not the case, the value of μ is increased and the process is repeated, until the desired relation holds.

For our benchmark, we use 9 different combinations of the values of n, m and m_1, and randomly generate 20 test functions for each combination. All components of the matrices A_j, vectors a_j and scalars c_j are chosen randomly within the interval $[-100, 500]$. The values of n, m and m_1 are reported in Table 1, as well as the average values of μ and the maximum number of iterations allowed in the corresponding runs.

Combination #	1	2	3	4	5	6	7	8	9
n	5	5	20	20	20	50	50	100	100
m	4	4	10	20	20	15	60	30	30
$m_1 = dim\,\mathcal{V}(p)$	3	1	3	3	15	8	8	5	25
μ	6129	5310	5456	13193	12642	15599	14067	18875	17187
`MaxBB`	100	100	300	300	300	400	400	400	400

Table 1: Some relevant data of the test-set

In our runs, only active bundle elements are kept at each iteration. Hence, in (3.3) the cutting-planes model is defined by taking the maximum of the affine functions for $i \in \tilde{I}_{k-1}$. The quadratic program (3.2) is solved by the built-in MATLAB QP solver.

4.2 Assessing solution quality

In order to determine the quality of the $\mathcal{VU}$-subspaces obtained from the objects computed by the bundle technique, we use four different criteria.

4.2.1 Prox accuracy.

The first measure is the one from [8], and is based on the knowledge of the exact proximal point $p = p_\mu(x^0)$. More precisely, let x^{best} denote the point triggering the stopping test of the analyzed variant at the iteration $k$best. Then the formula

$$AC := -\log_{10}\left(\frac{\|x^{\text{best}} - p\|}{1 + \|p\|}\right) \tag{4.3}$$

measures the accuracy in computing the actual proximal point. (Adding the term 1 to $\|p\|$ in the numerator of (4.3) measures the absolute accuracy when $\|p\|$ is small and the relative accuracy otherwise – in our case p is close to $\bar{x} = 0$). On this semi-log scale, a positive number (roughly) represents the number of digits of accuracy obtained with the variant.

4.2.2 $\mathcal{V}$-approximation

The next two measures estimate the quality of the estimated $\mathcal{V}$-subspace, where we use the short notation $\mathcal{V}$ to refer to $\mathcal{V}(p)$. An important point arises in relation to the amount of knowledge made available by the black-box. For the `max-quad` family, the black-box (3.1) gives as its output an index $j_i \in \{0, 1, \ldots, m\}$, corresponding to some structure function yielding the maximum.[2]

As a result, knowing the structure functions f_j, after each call to the black-box

- the function value $f(x^i)$, and
- a subgradient $\gamma^i \in \partial f(x^i)$

are available, via the relations $f(x^i) = f_{j_i}(x^i)$ and $\gamma^i = f'_{j_i}(x^i)$. In fact, the cutting-planes models (3.3) can be built solely based on the pairs $(f(x^i), \gamma^i)$. No additional knowledge (such as the identities $f(x^i) = f_{j_i}(x^i)$ and $\gamma^i = f'_{j_i}(x^i) = A_{j_i}x^i + a_{j_i}$, or the number of active structure functions $m_1 + 1$) is used by the algorithm to define the iterates in (3.2). The actual knowledge of the different indices j_i is used only to approximate the vectors v_l/s_l, $l = 0, 1, \ldots, m_1$, (in the notation of Theorem 3.2) that estimate a basis for $\mathcal{V}$. Specifically, keeping in the bundle memory the structure indices j_i allows to build the vectors

$$w_l^k := \frac{\sum_{i \in \tilde{I}_k;\, j_i = l} t_i^k f'_{j_i}(x^i)}{\sum_{i \in \tilde{I}_k;\, j_i = l} t_i^k} \tag{4.4}$$

that asymptotically tend to v_l/s_l, $l = 0, 1, \ldots, m_1$.

For other classes of functions, however, the corresponding black-box may only provide $f(x^i)$ and $\gamma^i \in \partial f(x^i)$, but not the index j_i. In such cases, when full structure knowledge is not available, one can still build bundle iterates $\{x^k\}$ that approximate the proximal point. Whether or not one can still approximate the $\mathcal{V}$-space remains an open theoretical question. A purely practical answer to this question would be to estimate $\mathcal{V}$ by replacing the unknown vectors w_l^k by the subgradients γ^i provided by the black-box. Note that, by (3.17), for k sufficiently large the accumulated multipliers q_l^k are all positive. This means, in particular, that for each $l \in \{0, 1, \ldots, m_1\}$ there is a bundle index i_l such that $t_{i_l}^k > 0$ and $f'_{j_{i_l}}(x^{i_l}) = f'_l(x^{i_l})$. Hence, from some iteration on, all the relevant structure gradients are present in the bundle. This remark justifies the variant $\mathcal{V}_\gamma$ below as an alternative approach, reasonable but heuristic, to estimate the $\mathcal{V}$-subspace.

We outline next two alternative approaches to compute $\mathcal{V}$, called $\mathcal{V}_w$ and $\mathcal{V}_\gamma$. We emphasize that the developed theory covers the $\mathcal{V}_w$ variant, while $\mathcal{V}_\gamma$ is a heuristic.

At the final iteration $k$best, the final active bundle indices $\tilde{I}_{k\text{best}}$, defining $g^{k\text{best}}$ in (3.5), are available. The two alternative variants are the following.

$\mathcal{V}_w$: Compute the vectors $w_l^{k\text{best}}$, given by (4.4), for all $l \in L_{\text{best}}$, where

$$L_{\text{best}} := \left\{ l \in \{0, 1, \ldots, m_1\} : \exists i \in \tilde{I}_{k\text{best}} \text{ for which } j_i = l \right\}.$$

Take $l_1 \in L_{\text{best}}$ and form a matrix V_w^{best} such that

$$\text{the columns of } V_w^{\text{best}} \text{span the space } \operatorname{lin} \left\{ w_l^{k\text{best}} - w_{l_1}^{k\text{best}},\ l \in L_{\text{best}} \setminus \{l_1\} \right\}.$$

[2] In our runs, iterates stay in the ball $B_\varepsilon(\bar{x})$ defined in Proposition 3.1.1, so in fact the output indices belong to the subset $\{0, 1, \ldots, m_1\}$.

$\mathcal{V}_\gamma$: Take $i_1 \in \tilde{I}_{k\text{best}}$ and form a matrix V_γ^{best} such that

$$\text{the columns of } V_\gamma^{\text{best}}\text{span the space lin}\left\{\gamma^i - \gamma^{i_1},\ i \in \tilde{I}_{k\text{best}} \setminus \{i_1\}\right\}.$$

Matrices V^{best} correspond to a basis of the subspace $\mathcal{V}(x^{\text{best}})$, and the respective rank gives the approximate $\mathcal{V}$ dimension. As for the $\mathcal{U}$-subspace, in both cases we take

$$U^{\text{best}} = (V^{\text{best}})^{\perp}.$$

Accordingly, the $\mathcal{U}$-component of g^k is given by $g^k_{\mathcal{U}} = (U^{\text{best}})^{\top} g^k$, with $k = k\text{best}$.

Our second measure of the quality of approximation computes the relative error in the $\mathcal{V}$-dimension:

$$RE := \frac{\dim \mathcal{V} - \dim \mathcal{V}(x^{\text{best}})}{\dim \mathcal{V}}. \tag{4.5}$$

Note that a negative (respectively, positive) value of RE indicates an under (respectively, over) estimation of the exact $\mathcal{V}$-dimension.

A third measure, computed only if $\dim \mathcal{V} = \dim \mathcal{V}(x^{\text{best}})$, refers to the orthogonality of the relevant subspaces, in terms of absolute errors:

$$AE := \max\Big(\|V^\top U^{\text{best}}\|, \|U^\top V^{\text{best}}\|, \|U^{\text{best}\top} V\|, \|V^{\text{best}\top} U\|\Big), \tag{4.6}$$

where the matrices V and U represent the exact subspaces $\mathcal{V}$ and $\mathcal{U}$, respectively. Specifically,

$$\text{the columns of } V \text{ span the space lin}\left\{f'_j(p) - f'_0(p),\ j = 1, \ldots, m_1\right\},$$

and $U = V^{\perp}$.

4.2.3 Quality of bundle approximation

We can also check closeness of variants $\mathcal{V}_\gamma$ and $\mathcal{V}_w$, by measuring how well the final active bundle subgradients γ^i, $i \in \tilde{I}_{k\text{best}}$, approximate the accumulated vectors $w_l^{k\text{best}}$. At the final iteration, the accumulated multipliers $q_l^{k\text{best}}$, $l \in L_{\text{best}}$, are compared with the convex coefficients solving the linear system

$$\begin{bmatrix} f'_0(p) & \ldots & f'_{m_1}(p) \\ 1 & \ldots & 1 \end{bmatrix} \bar{q} = \begin{pmatrix} \mu(x^0 - p) \\ 1 \end{pmatrix}.$$

The corresponding measure is

$$CF := \|q^{k\text{best}} - \bar{q}\|_\infty. \tag{4.7}$$

Finally, when the stopping test is triggered we also count those indices of relevant structure functions that are absent in the final active bundle:

$$HM := \left|\{0, 1, \ldots, m_1\} \setminus L_{\text{best}}\right|. \tag{4.8}$$

For the approximation to be good, we expect these two last measures to be small or null: $CF \approx 0$ and $HM = 0$.

4.3 Variants composing the benchmark

We consider four variants of rules for stopping iterations.

Serious-step stopping test. This is the classical descent test in bundle methods, which stops iterations of approximating the proximal point $p_\mu(x^0)$ once sufficient decrease with respect to $f(x^0)$ is achieved:

$$f(x^k) - f(x^0) \leq \tilde{\sigma} \left(f(x^0) - \psi_{k-1}(x^k) \right).$$

In our experiments, we take $\tilde{\sigma} = 0.99$ (bundle methods usually employ smaller values for this Armijo-like parameter, for example $\tilde{\sigma} = 0.1$; we set a higher value here to strengthen the test). We refer to this variant as **Ser99**. At the final iteration, the $\mathcal{V}$ subspace is estimated with variant $\mathcal{V}_w$.

$\mathcal{VU}$-stopping tests. Similarly to [22], the iteration process stops when

$$f(x^k) - \psi_{k-1}(x^k) \leq \sigma \|g_{\mathcal{U}}^k\|^2,$$

for some $\sigma \in (0,1)$. For $\sigma = 10^{-4}$, we use $\mathcal{U}_w$ and $\mathcal{U}_\gamma$ and refer to the respective variants as **VU-w** and **VU-γ**. At the final iteration, the $\mathcal{V}$ subspace is estimated with the corresponding variant, namely $\mathcal{V}_w$ or $\mathcal{V}_\gamma$.

MaxBB-stopping test. The iteration process stops after the black-box was called $k =$`MaxBB` times, with `MaxBB` given in Table 1. We shall refer to this variant as **MaxBB**. At iteration `MaxBB`, the $\mathcal{V}$ subspace is estimated with variant $\mathcal{V}_w$.

The last stopping rule, in particular, is meant to test the asymptotic convergence result in Theorem 3.2. As such, it is expected to give the best performances.

4.4 Tables and Performance Profiles

Using formulæ (4.3)-(4.8), we calculated the corresponding measures for each test run on all the variants. In Table 2 we report the smallest, the mean and the largest values obtained for measures (4.3), (4.5), and (4.6) (for each test set and each variant), as well as the average

	AC			RE			AE			BB	Bad
Variant	min	mean	max	min	mean	max	min	mean	max	mean	Run
Ser99	0.00	2.61	6.16	0.00	0.72	0.96	0.00	0.00	0.07	2	160
VU-γ	6.11	8.15	10.25	0.00	0.00	0.00	0.00	0.00	0.00	44	15
VU-w	6.11	8.15	10.25	0.00	0.00	0.00	0.00	0.00	0.00	44	15
MaxBB	6.11	8.43	10.26	0.00	0.00	0.00	0.00	0.00	0.00	300	0

Table 2: Accuracy (AC) obtained in Prox, Relative Error (RE) in $\mathcal{V}$-dimension, Absolute Error (AE) in $\mathcal{VU}$-orthogonality, number of Black-Box (BB) calls, and Failures.

number of calls of the black-box ("BB mean") and the number of failures ("Bad Run"), either by false positives -finding a wrong $\mathcal{V}$-dimension after triggering the stopping test-, or by reaching the maximum of iterations.

In terms of accuracy of the approximate proximal point, results are very good for **VU-**γ and **VU-**w. As expected, the highest accuracy was obtained with the **MaxBB** variant, reflecting the asymptotic result stated in Theorem 3.2. By contrast, the **Ser99** variant always stopped at the second iteration and gave rather poor performances in all runs (we had set a minimum of 2 iterations at each run, and this stopping test was always triggered at this minimum). Since in all runs the results obtained with **Ser99** correspond to x^2, the corresponding mean accuracy can be taken as an indication of the average number of exact digits already present in the starting point x^0.

In terms of quality of the $\mathcal{V}$-approximation, **Ser99** failed 160 out of 180 cases in finding the exact $\mathcal{V}$-dimension. The remaining variants, by contrast, exhibit a high level of precision, and at least for these runs, both $\mathcal{VU}$-stopping tests seem to offer a good compromise between number of calls to the black-box and accurate estimation of the $\mathcal{V}$-subspace. In fact, both variants **VU-**γ and **VU-**w gave practically identical results. Over the 180 runs, they always stopped at the same iteration, differing only in the $\mathcal{V}$-basis estimation. Both variants stop after an average of 44 iterations, having found the exact dimension of the subspace $\mathcal{V} = \mathcal{V}(p)$, with very similar bases V_w^{best}, V_γ^{best}. An explanation for the practically identical behaviour of variants **VU-**w and **VU-**γ is that, in our experiments, at the final iteration we have that $|\tilde{I}_{k\text{best}}| = dim\,\mathcal{V}{+}1$, and for each $l = 0, \ldots, m_1$,

$$\{i \in \tilde{I}_{k\text{best}} : j_i = l\} = \{i_l\} \quad \Longrightarrow \quad q_l^{k\text{best}} = t_{i_l}^{k\text{best}} \text{ and } w_l^{k\text{best}} = f'_{j_{i_l}}(x^{i_l}).$$

In other words, the active bundle information proves to be rather economical, keeping only one structure gradient per index $l = 0, \ldots, m_1$.

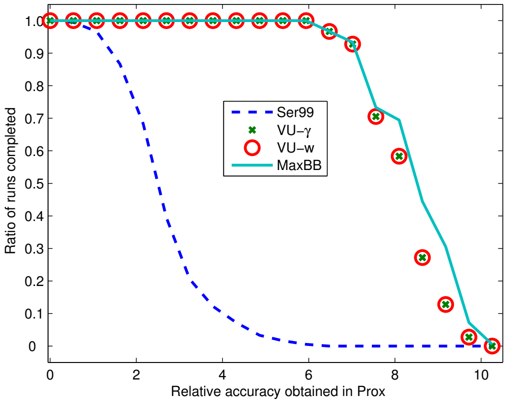

Figure 1: Performance Profile of prox-accuracy

To show in a graphical manner the degree of precision obtained by each variant, we also present some performance profiles. The performance profile in Figure 1 uses the scale (4.3) to plot the AC value on the x-axis versus the portion of tests which successfully achieved this value on the y-axis. Hence, the location where a profile first decreases from the y value 1 describes the gain in accuracy the variant achieved on every problem, while the location where a profile first obtains a y value of 0 yields the best gain in accuracy achieved using that variant. More generally, variants whose profiles are "higher" have out-performed algorithms with "lower" profiles.

We see in Figure 1 that both **VU** variants obtained at least 6 digits of accuracy in all the runs. Since starting points were taken "close enough" for our (local) results to hold, we analyze *a posteriori* the initial distance to the smooth manifold $\mathcal{M}$. Thus, for each one of the 180 starting points, we checked how many structure functions were active. We observed that only 3 starting points satisfied $I(x^0) = I(p)$, namely 2 and 1 starting points in combinations # 2 and # 3, respectively.

To determine the impact of the locality ball $B_\varepsilon(\bar{x})$, we made an additional test, performing again 180 runs with the same functions, this time eliminating the checking of closeness of x^0 to p. Since by (4.2), $\|x^0 - p\| = \gamma/\mu$, for the same γ and p considered in each one of the first 180 runs, we set $\mu = 1$ to "push" the new starting point away from the smooth manifold (in about a factor 10000 with respect to the previous runs, *cf.* Table 1). The corresponding results are highly instructive: out of the 180 functions considered, variant **VU-w** succeeded finding the exact $\mathcal{V}$-dimension for only 4 cases. Moreover, after having spent an average of 185 calls to the black-box, neither of the **VU** variants reached more than two digits of accuracy in the prox calculation.

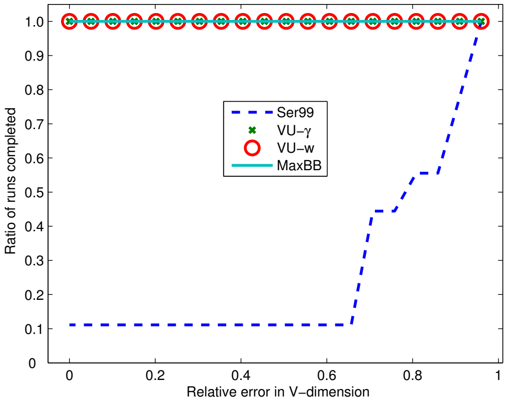

Figure 2: Performance Profile of relative error in $\mathcal{V}$-dimension

With respect to the relative error in $\mathcal{V}$-dimension, measured by the RE quotient in (4.5), we observe in the performance profile in Figure 2 that variant **Ser99** found the exact dimension in about 10% of the runs, while the **VU** variants succeeded in all the cases. This last result, in particular, means that the "Bad run" column in Table 2, showing 15 failures for the **VU** variants, corresponds to failures in triggering the stopping test before reaching `MaxBB`, and not to failures in finding the exact $\mathcal{V}$-dimension.

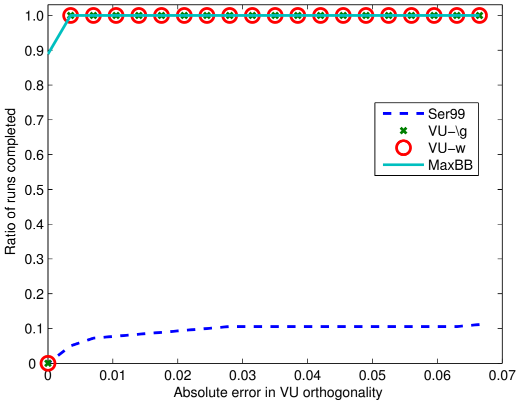

Figure 3: Performance Profile of absolute error in $\mathcal{VU}$-orthogonality

Figure 3 assesses once more the excellent performance of the **VU**-variant. For this Performance Profile, only successful runs (for which $RE = 0$) were considered, and this is the reason why 0.11 ($\approx (180-160)/180$) instead of 1 is the maximum value reached by the **Ser99** variant in the y axis.

Results using measures (4.7) and (4.8) again show that **VU-**γ and **VU-**w are comparable. In all the 180 cases $HM = 0$, with an average value of $CF = 0.02$.

Since the bundle scheme incorporates the knowledge of only 1 subgradient at each iteration, at the very least $1 + dim\,\mathcal{V}$ iterations are needed for having any hope of getting the right $\mathcal{V}$-dimension. We computed the relation between the number of calls to the black-box and the dimension of the $\mathcal{V}$-space, again averaging only over the cases where the stopping test was triggered and the right $\mathcal{V}$-dimension was found:

$$\frac{BBcalls - 1 - dim\,\mathcal{V}}{1 + dim\,\mathcal{V}}.$$

We found that **VU-**w needed in average 8 iterations per structure gradient, so higher $\mathcal{V}$-dimensions may require a high number of iterations to yield satisfactory estimations. It could then be thought that for higher dimensions a standard bundle method, like the code N1CV2 derived from [14], might be preferable. The computational work of N1CV2 per iteration is comparable to the variant **Ser99** (is actually slightly cheaper, as **Ser99** has the additional linear algebra calculations to compute the matrices V^{best} and U^{best}). A $\mathcal{VU}$-method, like the one in [22], is more expensive, as it solves a second quadratic programming problem per iteration to make the $\mathcal{U}$-Newton step. In spite of this apparent handicap, it is important to keep in mind the fact that standard bundle solvers tend to exhibit slow (linear or even sublinear) rate of convergence when approaching the solution. A graph comparing the superlinear convergence of the $\mathcal{VU}$-algorithm with the sublinear convergence of N1CV2 (which is one of the most efficient bundle solvers) can be found in [22, Figure 1]. When high precision is of interest, instead of making insignificant progress with many cheap iterations, it is preferable to spend more effort per iteration to progress fast, using the $\mathcal{VU}$-approach. Also, when close to a solution, the accumulated bundle is usually rich enough to allow good approximation of the proximal point, needed in the $\mathcal{VU}$-approach, at a reasonable price.

References

[1] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects.* Springer–Verlag, Berlin, Germany, 2006. Second Edition.

[2] X. Chen and M. Fukushima. Proximal quasi-Newton methods for nondifferentiable convex optimization. *Mathematical Programming*, 85:313–334, 1999.

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

[4] A. Daniilidis, W. Hare and J. Malick. Geometrical interpretation of the predictor-corrector type algorithms in structured optimization problems. *Optimization*, 55:481–503, 2006.

[5] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[6] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization* 17:259–286, 2006.

[7] W. Hare. *Nonsmooth optimization with smooth substructure.* PhD thesis, Department of Mathematics, Simon Fraser University, Canada, 2003.

[8] W. Hare and C. Sagastizábal. Benchmark of some nonsmooth optimization solvers for computing nonconvex proximal points. *Pacific Journal on Optimization*, 3:545–573, 2006.

[9] W. Hare and C. Sagastizábal. Computing proximal points of nonconvex functions. *Mathematical Programming*, 116:221–258, 2009.

[10] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms.* Springer–Verlag, Berlin, Germany, 1993.

[11] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization.* Lecture Notes in Mathematics, Vol. 1133. Springer–Verlag, Berlin, Germany, 1985.

[12] C. Lemaréchal, F. Oustry, and C. Sagastizábal. The $\mathcal{U}$-Lagrangian of a convex function. *Transactions of the American Mathematical Society*, 352:711–729, 2000.

[13] C. Lemaréchal and C. Sagastizábal. An approach to variable metric bundle methods. In J. Henry and J.-P. Yvon, editors, *Lecture notes in Control and Information Sciences No. 197, System Modelling and Optimization*, pages 144–162. Springer, Berlin, 1994.

[14] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods : From conceptual to implementable forms. *Mathematical Programming*, 76:393–410, 1997.

[15] A. Lewis. Active sets, nonsmoothness and sensitivity. *SIAM Journal on Optimization*, 13:702–725, 2002.

[16] J. Malick and S. Miller. Newton methods for nonsmooth convex minimization: connection among U-Lagrangian, Riemannian Newton and SQP methods. *Mathematical Programming*, 104:609–633, 2004.

[17] R. Mifflin. A quasi-second-order proximal bundle algorithm. *Mathematical Programming*, 73:51–72, 1996.

[18] R. Mifflin and C. Sagastizábal. On $\mathcal{VU}$-theory for functions with primal-dual gradient structure. *SIAM Journal on Optimization*, 11:547–571, 2000.

[19] R. Mifflin and C. Sagastizábal. Proximal points are on the fast track. *Journal of Convex Analysis*, 9:563–579, 2002.

[20] R. Mifflin and C. Sagastizábal. Primal-dual gradient structured functions: second-order results; links to epi-derivatives and partly smooth functions. *SIAM Journal on Optimization*, 13:1174–1194, 2003.

[21] R. Mifflin and C. Sagastizábal. $\mathcal{VU}$-Smoothness and Proximal Point Results for Some Nonconvex Functions. *Optimization Methods and Software*, 19:463–478, 2004.

[22] R. Mifflin and C. Sagastizábal. A $\mathcal{VU}$-algorithm for convex minimization. *Mathematical Programming*, 104:583–608, 2005.

[23] L. Qi and X. Chen. A preconditioning proximal Newton's method for nondifferentiable convex optimization. *Mathematical Programming*, 76:411–430, 1995.

[24] A. Shapiro. On a class of nonsmooth composite functions. *Mathematics of Operations Research*, 28:677–692, 2003.

[25] S.J. Wright. Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization*, 31:1063–1079, 1993.