Optimization: A Journal of Mathematical Programming and Operations Research

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gopt20

A splitting bundle approach for non-smooth non-convex minimization

A. Fuduli[a], M. Gaudioso[b] & E.A. Nurminski[c]

[a] Dipartimento di Matematica e Informatica, Università della Calabria, Rende, Italy.

[b] Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica - DIMES, Università della Calabria, Rende, Italy.

[c] Far Eastern Federal University and Institute for Automation and Control Problems Far Eastern Branch of Russian AS, Vladivostok, Russia.

Published online: 30 Sep 2013.

To cite this article: A. Fuduli, M. Gaudioso & E.A. Nurminski (2015) A splitting bundle approach for non-smooth non-convex minimization, Optimization: A Journal of Mathematical Programming and Operations Research, 64:5, 1131-1151, DOI: 10.1080/02331934.2013.840625

To link to this article: http://dx.doi.org/10.1080/02331934.2013.840625

Optimization, 2015
Vol. 64, No. 5, 1131–1151, http://dx.doi.org/10.1080/02331934.2013.840625

Taylor & Francis
Taylor & Francis Group

A splitting bundle approach for non-smooth non-convex minimization

A. Fuduli[a], M. Gaudioso[b]* and E.A. Nurminski[c]

[a]*Dipartimento di Matematica e Informatica,Università della Calabria, Rende, Italy;* [b]*Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica – DIMES, Università della Calabria, Rende, Italy;* [c]*Far Eastern Federal University and Institute for Automation and Control Problems Far Eastern Branch of Russian AS, Vladivostok, Russia*

(*Received 10 March 2013; accepted 11 August 2013*)

We present a bundle-type method for minimizing non-convex non-smooth functions. Our approach is based on the partition of the bundle into two sets, taking into account the local convex or concave behaviour of the objective function. Termination at a point satisfying an approximate stationarity condition is proved and numerical results are provided.

Keywords: non-smooth optimization; bundle methods; non-convex optimization

AMS Subject Classifications: 90C26; 65K05

1. Introduction

We tackle the following unconstrained minimization problem:

$$\min_{x\in\mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a possibly non-convex and not necessarily differentiable function.

In many practical applications, one is faced with the need of solving problems which are at the same time non-convex and non-smooth. Among the others, we cite here some recent applications in Machine Learning [1–6] where both non-smoothness [7] and non-convexity [8] enter into the play.

The literature on treatment of non-differentiability in convex and non-convex cases is extremely rich (see [9]). Some sample papers in the convex case are [10–19]. Moreover in [20–23] several techniques allowing inexact calculation of the objective function are introduced.

In the non-convex setting, many algorithms can be considered as the natural evolution of bundle-type methods [24,25] originally devised for dealing with convex minimization.[26] We recall here [27–33]. Different approaches are based on appropriate extensions of algorithms working for smooth problems. We cite [34,35], which modify the Newton or Quasi-Newton method to cope with non-smoothness. Gradient sampling and discrete gradient techniques have been fruitfully adopted in [36–39] and [40,41], respectively. Finally in [42,43] the authors present some techniques for the minimization of non-convex maximum

*Corresponding author. Email: gaudioso@deis.unical.it

eigenvalue functions and for non-smooth functions which are infinite maxima of eigenvalue functions.

The approach presented in this paper belongs to the bundle class and it is based on the construction of a piecewise affine model of the objective function. It is related to [44–46]. In particular, it shares with [45] the basic idea of partitioning the bundle of information into two subsets aimed at capturing, respectively, a kind of convex and concave behaviour around the current point in the iterative descent procedure.

The basic difference with respect to [45] is in use of the partitioned bundle in the construction of the objective function model: in fact in [45] the bundle splitting is embedded in a kind of implicit trust region fashion, whereas in this paper a penalty function approach is adopted.

The paper is organized as follows. In Section 2, we introduce our approach at the basis of the bundle penalty method, which is described in Section 3 and whose convergence to stationary points is proved in Section 4. The quadratic subprogramme, which is to be solved at each iteration, is discussed in Section 5. Finally, in Section 6, some numerical results are presented.

Throughout the paper, we denote by $x^T y$ the inner product of the vectors x and y.

2. The basic approach

We assume that f is locally Lipschitz, i.e. it is Lipschitz on every bounded set. Then, given a point x, the generalized gradient (or Clarke's gradient or subdifferential) is defined as follows:

$$\partial f(x) = \text{conv}\{g | g \in \mathbb{R}^n, \nabla f(x_k) \to g, x_k \to x, x_k \notin \Omega_f\}$$

where Ω_f is the set (of zero measure) where f is not differentiable. An extension of the generalized gradient is the *Goldstein ϵ-subdifferential* $\partial_\epsilon^G f(x)$ defined as

$$\partial_\epsilon^G f(x) = \text{conv}\{\partial f(y) | \|y - x\| \leq \epsilon\}.$$

We assume also that, at any point x, we are able to compute both the objective function value and a subgradient $g \in \partial f(x)$, i.e. an element of the generalized gradient.

Now we introduce our approach, recalling the basic bundle splitting idea of [45]. We denote by x_j the so-called 'stability centre', corresponding to the current estimate of a minimum in an iterative procedure, and by g_j any subgradient of f at x_j. At each iteration, the bundle of available information is the set

$$B \stackrel{\triangle}{=} \{(x_i, f(x_i), g_i, \alpha_i, a_i) | i \in I\}$$

where x_i , $i \in I$, are the iterates generated at the previous iterations, g_i is a subgradient of f at x_i, α_i is the linearization error between the actual value of the objective function at x_j and the linear expansion generated at x_i and evaluated at x_j, i.e.

$$\alpha_i \stackrel{\triangle}{=} f(x_j) - f(x_i) - g_i^T (x_j - x_i), \tag{2.1}$$

and

$$a_i \stackrel{\triangle}{=} \|x_j - x_i\|.$$

It is worth noting that, in the non-convex case, α_i may be negative, since the first order expansion at any point does not necessarily support from below the epigraph of the function.

As in [45], we partition the set I into two sets I_+ and I_- defined as follows

$$I_+ \triangleq \{i|\alpha_i \geq 0\} \quad I_- \triangleq \{i|\alpha_i < 0\}. \tag{2.2}$$

The partition of I induces the partition of B according to the index sets I_+ and I_-. The related points x_i can be interpreted as points exhibiting, respectively, a kind of 'convex behaviour' and 'concave behaviour' relatively to x_j. We observe that I_+ is never empty as at least the element $(x_j, f(x_j), g_j, 0, 0)$ belongs to the bundle.

We define the following piecewise affine functions:

$$f_+(x) \triangleq \max_{i \in I_+}\{f(x_i) + g_i^T(x - x_i)\} \tag{2.3}$$

and

$$f_-(x) \triangleq \max\left\{0, \max_{i \in I_-}[f(x_i) + g_i^T(x - x_i)]\right\}. \tag{2.4}$$

Indicating by d the 'displacement' from x_j, i.e. $d \triangleq x - x_j$, taking into account (2.1) and neglecting the constant term $f(x_j)$, from (2.3) and (2.4) we obtain, respectively, the following piecewise affine functions:

$$\Delta_+(d) \triangleq \max_{i \in I_+}\left\{g_i^T d - \alpha_i\right\}$$

and

$$\Delta_-(d) \triangleq \max\left\{0, \max_{i \in I_-}\left[g_i^T d - \alpha_i\right]\right\}.$$

Then in order to compute a tentative displacement we solve the following problem:

$$\min_d h(d), \tag{2.5}$$

with

$$h(d) \triangleq \frac{1}{2\gamma}\|d\|^2 + \Delta_+(d) + u\Delta_-(d), \tag{2.6}$$

where $\gamma > 0$ is the classic proximity parameter for bundle methods, u can be interpreted as a positive penalty parameter and $\|\cdot\|$ is the Euclidean norm.

Note that function Δ_+ corresponds to the classic 'cutting plane function', which is at the basis of the well-known cutting plane method.[47,48] At $d = 0$, while Δ_+ interpolates the difference function $f(x_j + d) - f(x_j)$ (since the index j belongs to I_+ and it is $\alpha_j = 0$), function Δ_- is strictly positive around $d = 0$, provided I_- is non-empty. Then the effect of adding Δ_- in the objective function h of problem (2.5) is to penalize the choice of 'small' displacements with respect to the current stability centre. Note also that h is strictly convex and admits unique minimum.

Problem (2.5) can be rewritten in the form of a quadratic programme as follows:

$$\begin{cases} \min\limits_{d,v,z} & q(d, v, z) \\ & v \geq g_i^T d - \alpha_i, \quad i \in I_+ \\ & z \geq g_i^T d - \alpha_i, \quad i \in I_- \\ & z \geq 0, \end{cases} \tag{2.7}$$

where

$$q(d, v, z) \stackrel{\triangle}{=} \frac{1}{2\gamma}\|d\|^2 + v + uz.$$

The dual of programme (2.7) is

$$\begin{cases} \min\limits_{\lambda\geq 0, \mu\geq 0} & \dfrac{\gamma}{2}\|G_+\lambda + G_-\mu\|^2 + \alpha_+^T\lambda + \alpha_-^T\mu \\ & e^T\lambda = 1 \\ & e^T\mu \leq u \end{cases} \tag{2.8}$$

where G_+ and G_- are matrices whose columns are, respectively, the vectors g_i, $i \in I_+$, and g_i, $i \in I_-$. The symbol e indicates a vector of ones of appropriate dimension. The terms α_i, $i \in I_+$ and α_i, $i \in I_-$, are grouped into the vectors α_+ and α_-, respectively.

The optimal primal solution $(d_{\gamma u}, v_{\gamma u}, z_{\gamma u})$ is related to the optimal dual solution $(\lambda_{\gamma u}, \mu_{\gamma u})$ by the following formulae:

$$d_{\gamma u} = -\gamma\left(G_+\lambda_{\gamma u} + G_-\mu_{\gamma u}\right) \tag{2.9a}$$

$$v_{\gamma u} + uz_{\gamma u} = -\frac{1}{\gamma}\|d_{\gamma u}\|^2 - \alpha_+^T\lambda_{\gamma u} - \alpha_-^T\mu_{\gamma u}. \tag{2.9b}$$

3. The algorithm

Our method is based on repeatedly solving problem (2.5). As in [45], by 'main iteration' we intend the set of steps where the stability centre remains unchanged. From the 'main iteration', two exits may occur:

(i) termination, due to the satisfaction of an approximate stationarity condition;
(ii) update of the stability centre, if sufficient decrease in the objective function is achieved.

The initialization of the algorithm requires a starting point $x_0 \in \mathbb{R}^n$ and the initial stability centre y coincides with x_0. The initial bundle is made up by just one element $(y, f(y), g_y, 0, 0)$, where $g_y \in \partial f(y)$, so that I_- is the empty set, while I_+ is a singleton. The following global parameters are to be set:

- the stationarity tolerance $\delta > 0$ and the proximity measure $\epsilon > 0$;
- the descent parameter $m \in (0, 1)$ and the cut parameter $\rho \in (m, 1)$;
- the increase parameter $R > 1$;
- the decrease parameter $r \in (0, 1)$;
- the threshold $\eta > 0$ on the expected reduction;
- the threshold $\beta > 0$ on the linearization errors;
- the penalty parameter $u > 0$ on function Δ_-.

A short description of the algorithm is the following:

ALGORITHMIC SCHEME

(1) Initialization.
(2) 'Main iteration'.
(3) Updating of the bundle of information w.r.t. the new stability centre.

In the sequel, we describe in detail the 'main iteration' without indexing it for sake of notational simplicity.

We remark that in general the 'main iteration' maintains the (updated) bundle of information from previous iterations. Updating the bundle is necessary since the quantities α_i and a_i are dependent on the stability centre.

ALGORITHM 3.1 (Main Iteration)

(0) *If* $\|g_y\| \leq \delta$ *then* STOP *(stationarity achieved), else set*

$$\bar{\gamma} := \frac{\sqrt{4\beta^2 u^2 + 4\|g_y\|^2\epsilon^2} - 2\beta u}{2\|g_y\|^2},$$

$$\gamma_{min} := r\bar{\gamma}, \quad \gamma_{max} := R\gamma_{min}, \quad \textit{and} \quad \theta := r\gamma_{min}\delta.$$

Select $\gamma \in [\gamma_{min}, \gamma_{max}]$.

(1) *Solve program (2.5), obtain* $d_{\gamma u}$ *and compute*

$$v_{\gamma u} = \max_{i \in I_+} \left\{ g_i^T d_{\gamma u} - \alpha_i \right\}.$$

If $\|d_{\gamma u}\| \leq \theta$ *then go to* 3, *else if* $v_{\gamma u} \leq -\eta$ *or* $I_- = \emptyset$ *then go to* 4.

(2) *Select an index* $i \in I_-$, *set* $I_- := I_- \setminus \{i\}$ *and go to* 1.

(3) *Set*

$$I_+ := I_+ \setminus \{i \in I_+ \mid a_i > \epsilon\}$$

and

$$I_- := \emptyset.$$

Calculate g^* *such that*

$$\|g^*\| = \min_{g \in \text{conv}\{g_i \mid i \in I_+\}} \|g\|.$$

If $\|g^*\| \leq \delta$ *then* STOP *(approximate stationarity achieved), else* $\gamma_{max} := \gamma_{max} - r(\gamma_{max} - \gamma_{min})$ *and go to* 1.

(4) *Set* $x := y + d_{\gamma u}$. *If*

$$f(x) \leq f(y) + m v_{\gamma u} \tag{3.1}$$

then set the new stability centre $y := x$ *and* EXIT *from the main iteration.*

(5) *Calculate* $g \in \partial f(x)$ *and set*

$$\alpha := \max\{-\beta, f(y) - f(x) + g^T d_{\gamma u}\}.$$

(a) *If* $\alpha < 0$ *and* $\|d_{\gamma u}\| > \epsilon$ *then insert the element* $(x, f(x), g, \alpha, \|d_{\gamma u}\|)$ *into the bundle for an appropriate value of* $i \in I_-$ *and set* $\gamma := \gamma - r(\gamma - \gamma_{min})$.

(b) *Else, if* $g^T d_{\gamma u} \geq \rho v_{\gamma u}$ *then insert the element* $(x, f(x), g, \max(0, \alpha), \|d_{\gamma u}\|)$ *into the bundle for an appropriate value of* $i \in I_+$.

(c) *Else find a scalar* $t \in (0, 1)$ *such that a subgradient* $g_t \in \partial f(y + t d_{\gamma u})$ *satisfies the condition* $g_t^T d_{\gamma u} \geq \rho v_{\gamma u}$ *and insert the element* $(y + t d_{\gamma u}, f(y + t d_{\gamma u}), g_t, \max(0, \alpha_t), t\|d_{\gamma u}\|)$ *into the bundle for an appropriate value of* $i \in I_+$, *where* $\alpha_t = f(y) - f(y + t d_{\gamma u}) + t g_t^T d_{\gamma u}$.

(6) *Go to* 1.

4. Convergence

In this section, we prove the finite termination of the overall method, under the following assumptions:

A1 f is weakly semismooth;

A2 the set $\mathcal{F}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is compact, with Lipschitz constant equal to L_0.

Although Algorithm 3.1 is explicitly based on repeatedly solving problem (2.5), we show the convergence by referring to iterative solutions of programme (2.7), which is equivalent to (2.5). Throughout the section we indicate by $(d_{\gamma u}, v_{\gamma u}, z_{\gamma u})$ and $(d_{\gamma u}, v_{\gamma u})$ the optimal solutions of programme (2.7), when $I_- \neq \emptyset$ and $I_- = \emptyset$, respectively. The corresponding optimal function value, for fixed positive values of γ and u, is indicated by $q_{\gamma u}$.

LEMMA 4.1 *For all* γ *and* u, *it holds*

$$\|d_{\gamma u}\| \leq \sqrt{\|g_y\|^2 \gamma^2 + 2\beta u \gamma} \qquad \text{if } I_- \neq \emptyset \tag{4.1}$$

and

$$\|d_{\gamma u}\| \leq \|g_y\| \gamma \qquad \text{if } I_- = \emptyset. \tag{4.2}$$

Proof Consider the case $I_- \neq \emptyset$ and let $(\lambda_{\gamma u}, \mu_{\gamma u})$ be the optimal solution to program (2.8). Take the feasible solution $(\bar{\lambda}, \bar{\mu})$ with $\bar{\mu} = 0$ and $\bar{\lambda}$ having all the components equal to zero, except the one in correspondence to $(y, f(y), g_y, 0, 0)$, which is set equal to 1. Then we have

$$\frac{\gamma}{2}\|G_+\lambda_{\gamma u} + G_-\mu_{\gamma u}\|^2 + \alpha_+^T\lambda_{\gamma u} + \alpha_-^T\mu_{\gamma u} \leq \frac{\gamma}{2}\|G_+\bar{\lambda} + G_-\bar{\mu}\|^2 + \alpha_+^T\bar{\lambda} + \alpha_-^T\bar{\mu},$$

i.e.

$$\frac{\gamma}{2}\|G_+\lambda_{\gamma u} + G_-\mu_{\gamma u}\|^2 + \alpha_+^T\lambda_{\gamma u} + \alpha_-^T\mu_{\gamma u} \leq \frac{\gamma}{2}\|g_y\|^2.$$

Thus, taking into account (2.9a), we obtain

$$\frac{1}{2\gamma}\|d_{\gamma u}\|^2 \leq \frac{\gamma}{2}\|g_y\|^2 - \alpha_-^T\mu_{\gamma u}. \tag{4.3}$$

Because $e^T\mu_{\gamma u} \leq u$ and $-\alpha_- \leq \beta e$, we have:

$$-\alpha_-^T\mu_{\gamma u} \leq \beta u. \tag{4.4}$$

Combining (4.3) and (4.4), we obtain

$$\|d_{\gamma u}\|^2 \leq \|g_y\|^2\gamma^2 + 2\beta u\gamma,$$

which completes the proof for the case $I_- \neq \emptyset$. The case $I_- = \emptyset$ can be easily proved in the same way, by considering that whenever $I_- = \emptyset$ the variable $\mu_{\gamma u}$ does not appear in the formulation of problem (2.8). □

LEMMA 4.2 *For all $u > 0$ there exists a positive value $\bar{\gamma}$ such that for $\gamma \in (0, \bar{\gamma}]$ it holds*

$$\|d_{\gamma u}\| \leq \epsilon.$$

Proof For $\beta > 0$ fix

$$\bar{\gamma} \triangleq \frac{\sqrt{4\beta^2u^2 + 4\|g_y\|^2\epsilon^2} - 2\beta u}{2\|g_y\|^2} > 0$$

and observe that, from $\delta < \|g_y\| \leq L_0$, one has that $\bar{\gamma}$ is bounded away from zero, since it is

$$\bar{\gamma} > \frac{\sqrt{4\beta^2u^2 + 4\delta^2\epsilon^2} - 2\beta u}{2L_0^2} > 0.$$

The property follows by simply substituting $\bar{\gamma}$ in (4.1) and (4.2) and taking into account that

$$\beta u\sqrt{4\beta^2u^2 + 4\|g_y\|^2\epsilon^2} \geq 2\beta^2u^2.$$

□

Remark 4.3 On the basis of Lemma 4.2, and taking into account $\gamma_{min} = r\bar{\gamma}$, with $r \in (0, 1)$, an infinite number of insertions of bundle indices into I_- cannot occur. In fact each time such an insertion takes place, γ is reduced and its updating formula ensures that, after a finite number of updates, γ becomes smaller than $\bar{\gamma}$ and, consequently, all the newly generated bundle indices are in I_+.

LEMMA 4.4 *Algorithm 3.1 cannot cycle infinitely many times through steps 1 and 3.*

Proof Assume by contradiction that such case occurs, that is the algorithm never stops for satisfaction of the criterion at step 3. Let us index by $k \in \mathcal{K}$ all the quantities referred to at the k-th passage. We have

$$\|d_{\gamma u}^{(k)}\| \leq \theta$$

and

$$\|g^{*(k)}\| > \delta.$$

Observe that $\gamma \leq \gamma_{max}$ and that by construction γ_{max} reduces in a finite number of steps below the threshold $\bar{\gamma}$. Thus, from Lemma 4.2, it follows that asymptotically $\|d_{\gamma u}^{(k)}\| \leq \epsilon$, which in turn implies that the indices of the new bundle elements are asymptotically inserted into I_+.

Because at step 3 we set $I_- = \emptyset$, from the above considerations and taking into account (2.9a) and the constraints in problem (2.8), it follows that there exists an index $\bar{k} \in \mathcal{K}$ such that for all $k \geq \bar{k}$ the direction $d_{\gamma u}^{(k)}$ can be expressed in the form

$$d_{\gamma u}^{(k)} = -\gamma g^{(k)},$$

with $g^{(k)} \in \text{conv}\{g_i \mid i \in I_+^{(k)}\}$. But since $\|d_{\gamma u}^{(k)}\| \leq \theta$ and $\|g^{*(k)}\| > \delta$, we have

$$\theta \geq \|d_{\gamma u}^{(k)}\| = \gamma \|g^{(k)}\| \geq \gamma_{min} \|g^{*(k)}\| > \frac{\theta}{r\delta}\delta = \frac{\theta}{r} > \theta,$$

reaching a contradiction. □

LEMMA 4.5 *For all γ and u it holds:*

(i) $\begin{array}{ll} -\gamma\|g_y\|^2 \leq q_{\gamma u} \leq u\beta & \text{if } I_- \neq \emptyset \\ -\gamma\|g_y\|^2 \leq q_{\gamma u} \leq 0 & \text{if } I_- = \emptyset. \end{array}$

(ii) $0 \leq z_{\gamma u} \leq \beta + \dfrac{\|g_y\|\sqrt{\|g_y\|^2\gamma^2 + 2\beta u\gamma}}{u}.$

Proof

(i) The triplet $(\bar{d}, \bar{v}, \bar{z}) = (0, 0, \beta)$ and the couple $(\bar{d}, \bar{v}) = (0, 0)$ are feasible for programme (2.7), respectively, in the two cases $I_- \neq \emptyset$ and $I_- = \emptyset$. The corresponding objective function values are $q(\bar{d}, \bar{v}, \bar{z}) = u\beta$ and $q(\bar{d}, \bar{v}) = 0$.
As for the lower bounds, note that, because the index corresponding to the bundle element $(y, f(y), g_y, 0, 0)$ belongs to I_+, q is minorized by the strictly convex function

$$\hat{q}(d) \stackrel{\triangle}{=} \frac{1}{2\gamma}\|d\|^2 + g_y^T d,$$

and the thesis follows taking into account that the minimal function value of $\hat{q}$ is $-\gamma\|g_y\|^2$.

(ii) From $q_{\gamma u} \leq u\beta$, from definition of v and taking into account that the element $(y, f(y), g_y, 0, 0)$ belongs to I_+, we have

$$\begin{aligned} uz_{\gamma u} &\leq u\beta - \frac{1}{2\gamma}\|d_{\gamma u}\|^2 - v_{\gamma u} \leq u\beta - v_{\gamma u} \\ &\leq u\beta + \|g_y\|\|d_{\gamma u}\| \leq u\beta + \|g_y\|\sqrt{\|g_y\|^2\gamma^2 + 2\beta u\gamma}, \end{aligned}$$

where the last inequality descends from Lemma 4.1. □

LEMMA 4.6 *Every time step 4 is entered we have*

$$|v_{\gamma u}| \geq \min\left\{\eta, \frac{\theta^2}{2\gamma}\right\}.$$

Proof At step 4, one arrives when $\|d_{\gamma u}\| > \theta$ and either $v_{\gamma u} \leq -\eta$ or $I_- = \emptyset$. The property is obviously true in case $v_{\gamma u} \leq -\eta$. If $I_- = \emptyset$, we have $q_{\gamma u} \leq 0$ and $v_{\gamma u} \leq 0$. Thus, from the definition of $q_{\gamma u}$, we obtain:

$$|v_{\gamma u}| \geq \frac{1}{2\gamma}\|d_{\gamma u}\|^2 > \frac{1}{2\gamma}\theta^2.$$

□

Remark 4.7 Consequence of Lemma 4.6 is that all times condition (3.1) is tested, $v_{\gamma u}$ is bounded away from zero. In fact θ depends on γ_{min}, while γ belongs to the interval $[\gamma_{min}, \gamma_{max}]$. Parameters γ_{min} and γ_{max} depend in turn on $\bar{\gamma}$, which is bounded away from zero (see Lemma 4.2).

LEMMA 4.8 *Let* $\left\{(d_{\gamma u}^{(k)}, v_{\gamma u}^{(k)}, z_{\gamma u}^{(k)})\right\}_{k\in\mathcal{K}}$ *be a subsequence generated within a single 'main iteration' such that*

$$\left\|d_{\gamma u}^{(k)}\right\| > \theta$$

and

$$f\left(y + d_{\gamma u}^{(k)}\right) - f(y) > m v_{\gamma u}^{(k)}$$

with the algorithm looping from step 1 *to step* 4. *Then the following hold:*

(i) *step* 5*(c) of the algorithm is well posed, i.e. there exist two nonnegative scalars* $t_1^{(k)}$ *and* $t_2^{(k)}$, $0 \leq t_1^{(k)} < t_2^{(k)} < 1$, *such that for any* $t \in [t_1^{(k)}, t_2^{(k)}]$ *the condition*

$$g(t)^T d_{\gamma u}^{(k)} \geq \rho v_{\gamma u}^{(k)}$$

is satisfied for every $g(t) \in \partial f(y + t d_{\gamma u}^{(k)})$.

(ii) *whenever a new bundle index is inserted into* I_+ *the condition*

$$g_k^T d_{\gamma u}^{(k)} \geq \rho v_{\gamma u}^{(k)}$$

holds, where g_k *is the subgradient corresponding to the new bundle element.*

Proof

(i) See proof of Lemma 4.1(ii) in [45]. Observe that at step 4 we arrive when $v_{\gamma u} \leq 0$.
(ii) See proof of Lemma 4.1(iii) in [45]. □

Remark 4.9 Property (i) of the above lemma guarantees well-posedness of step 5(c) in the sense that, letting ϵ_k be the length of the interval $[t_1^{(k)}, t_2^{(k)}]$, there exists a sufficiently large integer m (say $m \geq \frac{2}{\epsilon_k}$) such that an interval of length $\frac{1}{m}$ is contained in $[t_1^{(k)}, t_2^{(k)}]$. Consequently, sampling on all such intervals allows implementation of the step.

The proof of the following properties proceeds along guidelines which are similar to [45]. We report them for sake of completeness.

LEMMA 4.10 *Algorithm* 3.1 *cannot cycle infinitely many times through steps* 1 *and* 4.

Proof We need to show that it is impossible to have infinitely many times $\|d_{\gamma u}\| > \theta$ and the descent condition (3.1) not satisfied.

Indexing by $k \in \mathcal{K}$ the k-th passage through steps 1 and 4, we observe that, by Remark 4.3, there exists an index $\bar{k}$ such that for every $k \geq \bar{k}$ the index of each new bundle element is put in I_+ with γ and I_- remaining unchanged.

Under such condition, for $k \geq \bar{k}$ the sequence $\{q_{\gamma u}^{(k)}\}$ is monotonically non-decreasing, and, by Lemma 4.5, is bounded and hence it is convergent.

By Lemmas 4.1 and 4.5, respectively, the sequences $\{d_{\gamma u}^{(k)}\}$ and $\{z_{\gamma u}^{(k)}\}$ are bounded and admit a convergent subsequence, say $\{d_{\gamma u}^{(k)}\}_{k \in \mathcal{K}' \subseteq \mathcal{K}}$ and $\{z_{\gamma u}^{(k)}\}_{k \in \mathcal{K}' \subseteq \mathcal{K}}$, respectively.

The above considerations imply that also the sequence $\{v_{\gamma u}^{(k)}\}_{k \in \mathcal{K}' \subseteq \mathcal{K}}$ is convergent to a limit, say $\bar{v}$. Now let i and j be two successive indices in $\mathcal{K}'$ and $\zeta_i \stackrel{\triangle}{=} \max\{0, \alpha_i\}$, with $\alpha_i = \max\{-\beta, f(y) - f(y + d_{\gamma u}^{(i)}) + g_i^T d_{\gamma u}^{(i)}\}$ and $g_i \in \partial f(y + d_{\gamma u}^{(i)})$. We have

$$v_{\gamma u}^{(j)} \geq g_i^T d_{\gamma u}^{(j)} - \zeta_i, \tag{4.5}$$

$$f(y + d_{\gamma u}^{(i)}) - f(y) > m v_{\gamma u}^{(i)}$$

and, by Lemma 4.8,

$$g_i^T d_{\gamma u}^{(i)} \geq \rho v_{\gamma u}^{(i)}.$$

We note that

$$g_i^T d_{\gamma u}^{(i)} - \zeta_i \geq \rho v_{\gamma u}^{(i)}. \tag{4.6}$$

This inequality is trivially verified if $\zeta_i = 0$, and this occurs whenever it is $\alpha_i \leq 0$. Only the case $\alpha_i > 0$ is to be considered, that is $\alpha_i = f(y) - f(y + d_{\gamma u}^{(i)}) + g_i^T d_{\gamma u}^{(i)} > 0$. In fact, taking into account that $\rho > m$, it holds

$$g_i^T d_{\gamma u}^{(i)} - \zeta_i = f\left(y + d_{\gamma u}^{(i)}\right) - f(y) > m v_{\gamma u}^{(i)} > \rho v_{\gamma u}^{(i)}.$$

Combining (4.5) and (4.6) we obtain

$$v_{\gamma u}^{(j)} - \rho v_{\gamma u}^{(i)} \geq g_i^T \left(d_{\gamma u}^{(j)} - d_{\gamma u}^{(i)}\right)$$

and passing to the limit

$$(1 - \rho)\bar{v} \geq 0. \tag{4.7}$$

If $I_- \neq \emptyset$, inequality (4.7) is a contradiction because $\bar{v} \leq -\eta$. In case $I_- = \emptyset$, $\bar{v} \leq 0$ and inequality (4.7) implies $\bar{v} = 0$, which contradicts Lemma 4.6. □

From Lemmas 4.4 and 4.10, the following theorem descends.

THEOREM 4.11 *The* 'main iteration' *terminates after a finite number of steps.*

THEOREM 4.12 *For any $\epsilon > 0$ and $\delta > 0$, the algorithm stops in a finite number of 'main iterations' at a point satisfying the approximate stationarity condition*

$$\|g^*\| \leq \delta \text{with } g^* \in \partial_\epsilon^G f(y). \tag{4.8}$$

Proof The approximate stationarity condition (4.8) is exactly the stopping condition tested at step 3 of the 'main iteration'. Now suppose that it is not verified for an infinite number of 'main iteration' executions. From Theorem 4.11, it follows that infinitely many times the

descent condition is satisfied. Let $y^{(k)}$ be the stability centre at k-th passage through 'main iteration'; then $\|d_{\gamma u}^{(k)}\| > \theta^{(k)}$,

$$f(y^{(k+1)}) \leq f(y^{(k)}) + m v_{\gamma u}^{(k)}$$

and

$$f(y^{(k+1)}) - f(y^{(0)}) \leq m \sum_{i=0}^{k} v_{\gamma u}^{(i)}.$$

By Remark 4.7, $v_{\gamma u}^{(i)}$ is bounded away from zero. Therefore, by passing to the limit we obtain

$$\lim_{k\to\infty} f(y^{(k+1)}) - f(y^{(0)}) \leq -\infty$$

which is a contradiction, since f is bounded from below as a consequence of assumptions **A1** and **A2**. □

5. Computing the search direction

In this section, we focus on solving problem (2.5). We will show that such a problem reduces to finding a minimum norm vector inside a set given by the sum of polyhedra.

We use the following notation. Given a set A, we indicate by σ_A the support function of A, i.e.

$$\sigma_A(x) \stackrel{\triangle}{=} \max_{a\in A} a^T x$$

and by $\mathrm{Nr}(A)$ the minimum norm vector in A. Moreover $\mathrm{conv}(A)$ and $\mathrm{Co}(A)$ denote, respectively, the convex and the conic hulls of A.

Solving problem (2.5) is equivalent to

$$\min_d \bar{h}(d), \tag{5.1}$$

where

$$\bar{h}(d) \stackrel{\triangle}{=} \frac{1}{2}\|d\|^2 + \gamma \max_{i\in I_+} \left\{ g_i^T d - \alpha_i \right\} + \gamma_- \max \left\{ 0, \max_{i\in I_-} [g_i^T d - \alpha_i] \right\}, \tag{5.2}$$

with $\gamma_- \stackrel{\triangle}{=} u\gamma > 0$. Function $\bar{h}$ can be put in the form

$$\bar{h}(d) = \frac{1}{2}\|d\|^2 + \gamma \max_{b\in S_+} \sum_{i\in I_+} b_i (g_i^T d - \alpha_i) + \gamma_- \max_{c\in S_-} \sum_{j\in I_-} c_j \left(g_j^T d - \alpha_j \right),$$

where

$$S_+ \stackrel{\triangle}{=} \left\{ b \in \mathbb{R}^{|I_+|} | \sum_{i\in I_+} b_i = 1, \quad b_i \geq 0, i \in I_+ \right\}$$

and

$$S_- \stackrel{\triangle}{=} \left\{ c \in \mathbb{R}^{|I_-|} | \sum_{i\in I_-} c_i \leq 1, \quad c_i \geq 0, i \in I_- \right\}.$$

Letting

$$\bar{d} = \begin{bmatrix} d \\ 1 \end{bmatrix} \quad \text{and} \quad \bar{g}_i = \begin{bmatrix} g_i \\ -\alpha_i \end{bmatrix}$$

and indicating by $\bar{G}_+ \stackrel{\triangle}{=} \text{conv}\{\bar{g}_i, i \in I_+\}$ and $\bar{G}_- \stackrel{\triangle}{=} \text{conv}\{0, \bar{g}_i, i \in I_-\}$, function $\bar{h}$ becomes:

$$\begin{aligned}\bar{h}(\bar{d}) &= \frac{1}{2}\|\bar{d}\|^2 - \frac{1}{2} + \gamma \max_{b\in S_+} \sum_{i\in I_+} b_i \bar{g}_i^T \bar{d} + \gamma_- \max_{c\in S_-} \sum_{i\in I_-} c_i \bar{g}_i^T \bar{d} \\ &= \frac{1}{2}\|\bar{d}\|^2 - \frac{1}{2} + \sigma_{\gamma\bar{G}_+}(\bar{d}) + \sigma_{\gamma_-\bar{G}_-}(\bar{d}) \\ &= \frac{1}{2}\|\bar{d}\|^2 - \frac{1}{2} + \sigma_{(\gamma\bar{G}_+ + \gamma_-\bar{G}_-)}(\bar{d}).\end{aligned}$$

Thus problem (2.7) reduces to the following:

$$\begin{cases} -\dfrac{1}{2} + \min\limits_{\bar{d}} \dfrac{1}{2}\|\bar{d}\|^2 + \sigma_{(\gamma\bar{G}_+ + \gamma_-\bar{G}_-)}(\bar{d}) \\ e_{n+1}^T \bar{d} = 1, \end{cases} \tag{5.3}$$

whose Lagrangean dual is the one-dimensional problem

$$-\frac{1}{2} - \min_p\{p + \phi(p)\}, \tag{5.4}$$

where p is the dual variable corresponding to the constraint $e_{n+1}^T\bar{d} = 1$ and

$$\phi(p) \stackrel{\triangle}{=} -\min_{\bar{d}} \frac{1}{2}\|\bar{d}\|^2 + \sigma_{(\gamma\bar{G}_+ + \gamma_-\bar{G}_- + pe_{n+1})}(\bar{d}).$$

It can be shown that

$$\phi(p) = \frac{1}{2}\|\text{Nr}(\gamma\bar{G}_+ + \gamma_-\bar{G}_- + pe_{n+1})\|^2. \tag{5.5}$$

Note that the evaluation of the objective function ϕ of the univariate optimization problem (5.4) requires to solve a projection problem of the type (5.5). Note also that function ϕ is differentiable and, once $\phi(p)$ has been computed for a certain value of p, its derivative corresponds to the last component of the vector $\text{Nr}(\gamma\bar{G}_+ + \gamma_-\bar{G}_- + pe_{n+1})$. Similar problems have been treated in many papers. See [49–52].

Finally we remark that, in case γ_- is sufficiently large, the set $\gamma_-\bar{G}_-$ can be replaced by the cone $K_- = \text{Co}\{\bar{g}_i, i \in I_-\}$ and $\phi(p)$ simplifies to

$$\phi_{K_-}(p) \stackrel{\triangle}{=} \frac{1}{2}\|\text{Nr}(\gamma\bar{G}_+ + K_- + pe_{n+1})\|^2.$$

6. Numerical results

The algorithm described in Section 3 is not implementable as it requires, in principle, unbounded storage. A common way to overcome such difficulty is to introduce a subgradient aggregation technique (see [26,53]).

We proceed along the guidelines of [45]. In particular, let $(\hat{d}_{\gamma u}, \hat{v}_{\gamma u}, \hat{z}_{\gamma u})$ and $(\hat{\lambda}_{\gamma u}, \hat{\mu}_{\gamma u})$ be the optimal solutions of programmes (2.7) and (2.8), respectively, in correspondence to fixed values of γ and u. The aggregation is based on the following definitions:

$$g_{a_+} \stackrel{\triangle}{=} G_+\hat{\lambda}_{\gamma u}, \quad \alpha_{a_+} \stackrel{\triangle}{=} \alpha_+^T\hat{\lambda}_{\gamma u}$$

Table 1. L. Lukšan and J. Vlček test problems.

#	Problem	n	f^*
1	Rosenbrock	2	0
2	Crescent	2	0
3	CB2	2	1.9522245
4	CB3	2	2
5	DEM	2	−3
6	QL	2	7.2
7	LQ	2	−1.4142136
8	Mifflin1	2	−1
9	Mifflin2	2	−1
10	Rosen-Suzuki	4	−44
11	Shor	5	22.600162
12	Maxquad	10	−0.8414083
13	Maxq	20	0
14	Maxl	20	0
15	Goffin	50	0
16	El-Attar	6	0.5598131
17	Wolfe	2	−8
18	MXHILB	50	0
19	L1HILB	50	0
20*	Colville1	5	−32.348679
21	Gill	10	9.7857721
22*	TR48	48	−638565
23	Shell Dual	15	32.348679
24	Steiner2	12	16.703838

and, in case $I_- \neq \emptyset$,

$$g_{a_-} \triangleq \frac{G_- \hat{\mu}_{\gamma u}}{u}, \quad \alpha_{a_-} \triangleq \frac{\alpha_-^T \hat{\mu}_{\gamma u}}{u}.$$

The aggregate problem

$$\begin{cases} \min\limits_{d,v,z} \; q(d,v,z) \\ \\ v \geq g_{a_+}^T d - \alpha_{a_+} \\ \\ v \geq g_i^T d - \alpha_i, \quad i \in I_{a_+} \\ \\ z \geq g_{a_-}^T d - \alpha_{a_+} \\ \\ z \geq g_i^T d - \alpha_i, \quad i \in I_{a_-} \\ \\ z \geq 0, \end{cases} \tag{6.1}$$

has the same optimal solution $(\hat{d}_{\gamma u}, \hat{v}_{\gamma u}, \hat{z}_{\gamma u})$ as problem (2.7), where I_{a_+} and I_{a_-} are arbitrary subsets of I_+ and I_-, respectively.

Table 2. L. Lukšan and J. Vlček test problems: computational results.

	NCVX		NCVX$^{\text{penalty}}$ (10^{-6})			NCVX$^{\text{penalty}}$ (10^{-2})		
Problem	N_f	f	N_f	secs	f	N_f	secs	f
Rosenbrock	70	5.009E−07	72	1.98	1.338E−06	68	0.65	1.584E−05
Crescent	22	8.022E−06	27	0.73	7.550E−06	19	0.27	2.046E−05
CB2	18	1.9522245	20	0.58	1.9522246	19	0.21	1.9522249
CB3	15	2.0000001	19	0.64	2.0000000	19	0.26	2
DEM	21	−2.9999999	10	0.36	−3.0000000	15	0.26	−2.9999998
QL	28	7.2000005	21	0.62	7.2000001	21	0.49	7.2000001
LQ	9	−1.4142135	8	0.21	−1.4142136	9	0.15	−1.4142136
Mifflin1	93	−0.9999822	99	3.96	−0.9999839	104	0.7	−0.99995667
Mifflin2	13	−1.0000000	10	0.28	−0.9999999	12	0.24	−1.000000
Rosen	29	−44.0000000	36	1.49	−44.0000000	46	0.44	−43.999998
Shor	44	22.600162	48	1.83	22.600212	55	0.53	22.600163
Maxquad	56	−0.8414078	61	4.85	−0.8414077	65	0.82	−0.84140645
Maxq	293	1.660E−07	324	11.84	2.768E−05	303	5.58	1.174E−05
Maxl	44	1.110E−15	22	1.02	3.400E−07	43	0.9	6.392E−07
Goffin	148	1.142E−13	57	71.77	6.056E−8	59	34.38	3.452E−07
El-Attar	152	0.5598163	287	8.75	0.55981572	166	1.63	0.55981541
Wolfe	21	−7.9999998	26	1.07	−7.9999999	27	0.42	−8.0000000
MXHILB	33	1.768E−07	13	1.27	$2.936E-07$	14	0.84	$2.170E-07$
L1HILB	104	6.978E−07	33	9.04	8.468E−07	28	2.22	$7.523E-07$
Colville1*	47	−32.348679	51	1.43	–32.348678	73	0.56	−32.34866
Gill	164	9.7857746	253	32.05	9.7860516	169	1.39	9.7860738
TR48*	353	−638565	388	1340.7	–638565	397	1546.4	−638565
Shell Dual	1497	32.349404	1500	393.76	32.41996	1500	55.08	32.37687
Steiner2	196	16.703838	173	18.69	16.703839	171	2.7	16.703874

Note that monotonicity of the sequence $\{q_{\gamma u}^{(k)}\}$, necessary in the proof of Lemma 4.10, is guaranteed by the aggregation.

The algorithm, encompassing the aggregation scheme, has been implemented in double precision C++ under a *Linux Ubuntu* system.

The code, called NCVX$^{\text{penalty}}$, has been tested on two sets of functions. The first set, listed in Table 1 (see Lukšan and Vlček [54]), is constituted by 24 problems available on the web at the URL http://www.cs.cas.cz/~luksan/test.html. All test problems, except Rosenbrock, are non-smooth. We did not include function HS78 (reported in [54]), because it is unbounded from below and then does not satisfy assumption **A2**.

The second set, known as 'Ferrier polynomials',[29,55] is constituted by the following five test functions:

$$f_1(x) \stackrel{\triangle}{=} \sum_{i=1}^{n} |l_i(x)|$$

$$f_2(x) \stackrel{\triangle}{=} \sum_{i=1}^{n} (l_i(x))^2$$

Table 3. Ferrier polynomials: computational results.

		RedistProx		NCVX$^{\text{penalty}}$ (10^{-6})			NCVX$^{\text{penalty}}$ (10^{-2})		
k	n	N_f	f	N_f	secs	f	N_f	secs	f
1	1	2	0.000000	8	0.27	0.000000	8	0.16	0.000000
2	1	2	0.000000	1	0.00	0.000000	1	0.00	0.000000
3	1	2	0.000000	8	0.28	0.000000	8	0.18	0.000000
4	1	5	0.500000	12	0.42	0.500000	12	0.13	0.500000
5	1	10	0.000943	10	0.26	0.500000	10	0.15	0.500000
1	2	12	0.086533	22	0.57	2.758E−07	23	0.38	1.251E−06
2	2	16	0.000005	35	0.89	1.296E−06	35	0.32	9.789E−07
3	2	14	0.000000	8	0.27	1.621E−05	8	0.16	1.621E−05
4	2	10	0.036623	21	0.47	8.920E−07	21	0.23	8.942E−07
5	2	16	0.000000	26	0.74	2.450E−07	300	1.34	3.291E−06
1	3	13	0.000726	21	0.57	1.708E−08	22	0.31	2.284E−06
2	3	26	0.000000	17	0.2	1.922E−07	17	0.2	1.922E−07
3	3	14	0.000467	10	0.44	3.704E−06	10	0.16	3.704E−06
4	3	12	0.052922	30	0.9	4.402E−05	32	0.25	6.551E−07
5	3	16	0.000000	37	0.73	9.711E−07	29	0.29	4.923E−07
1	4	17	0.074170	32	0.89	1.292E−08	34	0.36	1.476E−08
2	4	18	0.000000	12	0.22	1.919E−11	12	0.2	1.919E−11
3	4	16	0.007471	11	0.4	2.819E−06	11	0.19	2.819E−06
4	4	18	0.025722	26	0.75	9.418E−07	24	0.32	6.072E−07
5	4	23	0.019105	47	0.77	2.272E−07	29	0.34	8.029E−07
1	5	23	0.213263	35	1.09	1.313E−07	51	0.47	8.579E−07
2	5	49	0.000000	23	2.93	1.007E−08	22	3.16	1.642E−07
3	5	20	0.680228	300	1.73	1.190E−06	300	1.37	1.160E−06
4	5	3	7.948708	40	1.02	3.014E−07	31	0.35	3.213E−07
5	5	26	0.352803	36	1.05	5.708E−07	43	0.37	8.689E−07
1	6	33	0.000436	44	1.6	7.200E−07	36	0.59	2.880E−07
2	6	35	0.000000	17	0.32	5.502E−07	17	0.21	5.498E−07
3	6	25	0.093031	54	0.89	9.200E−07	41	0.34	7.789E−07
4	6	41	0.022756	38	0.96	5.018E−08	33	0.31	7.836E−07
5	6	35	0.113835	52	1.04	3.001E−07	32	0.39	8.732E−08
1	7	34	0.146978	46	1.42	5.054E−07	78	0.6	1.268E−07
2*	7*	48	0.000000	–	–	–	–	–	–
3	7	30	0.018131	61	0.89	7.500E−07	61	0.4	7.497E−07
4	7	34	0.239281	39	1.36	2.323E−07	60	0.52	2.896E−08
5	7	41	0.120009	92	2.3	1.512E−07	81	0.57	1.461E−07
1	8	39	0.379768	163	5.49	1.752E−07	118	1.05	5.685E−07
2	8	59	0.074621	33	0.66	3.538E−08	33	0.38	3.539E−08
3	8	34	0.067895	84	1.17	6.957E−08	84	0.57	6.949E−08
4	8	32	0.797753	109	3.18	0.062617	70	0.64	0.0626197
5	8	38	1.077575	63	2.4	6.331E−07	89	0.72	7.895E−07
1	9	67	0.000001	168	8.54	4.820E−07	134	1.17	5.943E−07
2	9	37	0.000000	36	0.87	7.661E−09	38	0.38	1.801E−08
3	9	40	0.070714	83	1.09	6.742E−07	83	0.54	6.741E−07
4	9	61	0.000584	105	5.45	3.080E−07	105	1.09	7.576E−07
5	9	40	0.769331	160	5.83	7.533E−07	139	1.18	5.946E−07

(*continued*)

Table 3. (*Continued*).

		RedistProx		NCVX$^{\text{penalty}}$ (10^{-6})			NCVX$^{\text{penalty}}$ (10^{-2})		
k	n	N_f	f	N_f	secs	f	N_f	secs	f
1	10	52	0.106221	108	6.17	1.355E−06	115	1.36	4.389E−07
2*	10*	36	0.000000	–	–	–	36	0.41	3.226E−08
3	10	39	0.016641	66	0.95	1.014E−07	66	0.48	1.013E−07
4	10	46	0.165442	160	8.59	4.662E−07	115	0.96	3.806E−07
5	10	66	0.036327	129	5.53	8.415E−07	141	1.5	8.410E−07

Table 4. Ferrier polynomials: summary of Table 3.

	# wins with respect to N_f	# solved to $f^* < 10^{-6}$	# solved to $f^* < 10^{-3}$
RedistProx	37	13	20
NCVX$^{\text{penalty}}$	14	42	46

$$f_3(x) \stackrel{\triangle}{=} \max_{1\le 1\le n} |l_i(x)|$$

$$f_4(x) \stackrel{\triangle}{=} \sum_{i=1}^{n} |l_i(x)| + 0.5\|x\|^2$$

$$f_5(x) \stackrel{\triangle}{=} \sum_{i=1}^{n} |l_i(x)| + 0.5\|x\|,$$

where l_i is a real function of n real variables defined as follows:

$$l_i(x) \stackrel{\triangle}{=} \left(ix_i^2 - 2x_i - C\right) + \sum_{j=1}^{n} x_j,$$

with C being a fixed constant. All such test functions but f_2 are non-smooth; moreover if $C = 0$ then

$$\min_x f_k(x) = 0, \quad k = 1, \ldots, 5.$$

The parameters have been set as follows: $\delta = 10^{-4}$, $\epsilon = 10^{-2}$, $m = 0.2$, $r = 0.5$, $R = 10^6$, $\rho = 0.9$, $\eta = 0.1$, $\beta = 1$ and $u = 10^{-3}$. The maximum number of function evaluations has been fixed to 1500 for the first test set and 300 (as in [29]) for the second one.

We stop the code also when $I_- = \emptyset$ and $v_{\gamma u} \le \bar{\delta}$, with $\bar{\delta} = 10^{-6}$. This appears reasonable, because I_+ may contain also indices corresponding to negative linearization errors (see Step 5 of Algorithm 3.1) relative to points close to the stability centre.

The results of our numerical experiments on the L. Lukšan and J. Vlček test problems are reported in Table 2, where N_f indicates the number of function evaluations, *secs* is the

Table 5. Ferrier polynomials: Computational results.

		RedistProx		NCVX$^{\text{penalty}}$ (10^{-6})			NCVX$^{\text{penalty}}$ (10^{-2})		
k	n	N_f	f	N_f	secs	f	N_f	secs	f
1	1	2	0.000000	8	0.27	0.000000	8	0.16	0.000000
2	1	2	0.000000	1	0.00	0.000000	1	0.00	0.000000
3	1	2	0.000000	8	0.28	0.000000	8	0.18	0.000000
4	1	28	0.000000	12	0.42	0.500000	12	0.13	0.500000
5	1	13	0.000000	10	0.26	0.500000	10	0.15	0.500000
1	2	301	0.000031	22	0.57	2.758E−07	23	0.38	1.251E−06
2	2	22	0.000001	35	0.89	1.296E−06	35	0.32	9.789E−07
3	2	14	0.000000	8	0.27	1.621E−05	8	0.16	1.621E−05
4	2	301	0.000188	21	0.47	8.920E−07	21	0.23	8.942E−07
5	2	16	0.000000	26	0.74	2.450E−07	300	1.34	3.291E−06
1	3	19	0.000000	21	0.57	1.708E−08	22	0.31	2.284E−06
2	3	26	0.000000	17	0.2	1.922E−07	17	0.2	1.922E−07
3	3	86	0.000000	10	0.44	3.704E−06	10	0.16	3.704E−06
4	3	67	0.000001	30	0.9	4.402E−05	32	0.25	6.551E−07
5	3	16	0.000000	37	0.73	9.711E−07	29	0.29	4.923E−07
1	4	26	0.000000	32	0.89	1.292E−08	34	0.36	1.476E−08
2	4	18	0.000000	12	0.22	1.919E−11	12	0.2	1.919E−11
3	4	34	0.000002	11	0.4	2.819E−06	11	0.19	2.819E−06
4	4	301	0.000005	26	0.75	9.418E−07	24	0.32	6.072E−07
5	4	301	0.010121	47	0.77	2.272E−07	29	0.34	8.029E−07
1	5	149	0.000001	35	1.09	1.313E−07	51	0.47	8.579E−07
2	5	49	0.000000	23	2.93	1.007E−08	22	3.16	1.642E−07
3	5	261	0.000001	300	1.73	1.190E−06	300	1.37	1.160E−06
4	5	186	0.000036	40	1.02	3.014E−07	31	0.35	3.213E−07
5	5	91	0.002217	36	1.05	5.708E−07	43	0.37	8.689E−07
1	6	41	0.000000	44	1.6	7.200E−07	36	0.59	2.880E−07
2	6	35	0.000000	17	0.32	5.502E−07	17	0.21	5.498E−07
3	6	97	0.001095	54	0.89	9.200E−07	41	0.34	7.789E−07
4	6	60	0.000000	38	0.96	5.018E−08	33	0.31	7.836E−07
5	6	301	0.015945	52	1.04	3.001E−07	32	0.39	8.732E−08
1	7	301	0.000941	46	1.42	5.054E−07	78	0.6	1.268E−07
2*	7*	48	0.000000	–	–	–	–	–	–
3	7	38	0.000017	61	0.89	7.500E−07	61	0.4	7.497E−07
4	7	100	0.008193	39	1.36	2.323E−07	60	0.52	2.896E−08
5	7	67	0.018334	92	2.3	1.512E−07	81	0.57	1.461E−07
1	8	301	0.000008	163	5.49	1.752E−07	118	1.05	5.685E−07
2	8	168	0.000000	33	0.66	3.538E−08	33	0.38	3.539E−08
3	8	301	0.000024	84	1.17	6.957E−08	84	0.57	6.949E−08
4	8	301	0.000021	109	3.18	0.062617	70	0.64	0.0626197
5	8	301	0.000534	63	2.4	6.331E−07	89	0.72	7.895E−07
1	9	67	0.000001	168	8.54	4.820E−07	134	1.17	5.943E−07
2	9	37	0.000000	36	0.87	7.661E−09	38	0.38	1.801E−08
3	9	193	0.000010	83	1.09	6.742E−07	83	0.54	6.741E−07
4	9	150	0.000000	105	5.45	3.080E−07	105	1.09	7.576E−07
5	9	117	0.002150	160	5.83	7.533E−07	139	1.18	5.946E−07

(*continued*)

Table 5. (*Continued*).

		RedistProx		NCVX$^{\text{penalty}}$ (10^{-6})			NCVX$^{\text{penalty}}$ (10^{-2})		
k	n	N_f	f	N_f	secs	f	N_f	secs	f
1	10	85	0.000010	108	6.17	1.355E−06	115	1.36	4.389E−07
2*	10*	36	0.000000	–	–	–	36	0.41	3.226E−08
3	10	154	0.002658	66	0.95	1.014E−07	66	0.48	1.013E−07
4	10	67	0.002794	160	8.59	4.662E−07	115	0.96	3.806E−07
5	10	155	0.013286	129	5.53	8.415E−07	141	1.5	8.410E−07

Table 6. Ferrier polynomials: summary of Table 5.

	# wins with respect to N_f	# solved to $f^* < 10^{-6}$	# solved to $f^* < 10^{-3}$
RedistProx	17	22	40
NCVX$^{\text{penalty}}$	34	42	46

CPU time expressed in seconds and f indicates the function value reached by the algorithm when it stops. To compute the search direction, instead of solving directly problem (2.7) or (2.8), we solve problem (5.4) by means of a bisection technique. To evaluate function ϕ, we use the QP subroutine provided by the IBM ILOG CPLEX package (version 12.1).

Note that in our approach inexact solution of problem (5.4) is allowed. In fact all we need, at each iteration, is a primal feasible solution $(\bar{d}, \bar{v}, \bar{z})$ or $(\bar{d}, \bar{v})$ satisfying the conditions dictated by Lemma 4.1, Lemma 4.5 and such that $\bar{v} \leq 0$, whenever $I_- = \emptyset$. Then, in order to evaluate the behaviour of the code in terms of CPU time when problem (5.4) is solved with different tolerances, we have considered two cases: in the first one we stop the bisection procedure when $|\phi'| \leq 10^{-6}$, while in the second case when $|\phi'| \leq 10^{-2}$. We compare our results with those obtained by the NCVX code [45].

For the starred test functions Colville1 and TR48, we have adopted a different setting of some parameters, letting $\bar{\delta} = 10^{-5}$ for Colville1 and $m = 0.8$ for TR48. In Table 2, for each row the best N_f-value has been underlined. The comparison with NCVX code appears promising, because in about the half of the test functions (11 over 24), NCVX$^{\text{penalty}}$ performs better than NCVX in terms of number of function evaluations. As expected, solving approximately the Lagrangean problem (5.4) (bisection tolerance 10^{-2}) is in general faster than the 'exact' case (bisection tolerance 10^{-6}), giving sometimes better results also in terms of number of function evaluations.

As for the Ferrier polynomials test problems, we report our results in Tables 3 and 5 (for $n = 1, \ldots, 10$). We compare our method with the one described in [29], which is based on a local convexification of the objective function. The corresponding code is RedistProx and in [29] six different tables of results are presented, based on different combinations of the QP solver and of the bundle management adopted.

The comparison is made both in terms of number of function evaluations and precision. In particular, from among the six different implementations of [29], we report in Table 3 the best results obtained by RedistProx in terms of number of function evaluations, and in Table 5 the best results in terms of the objective function value. For each row the best results in terms of N_f-value and precision are underlined, respectively, in Tables 3 and 5.

Note that for the two starred problems ($k = 2$ and $n = 7, 10$) our code fails due to rounding errors in solving the QP subproblem.

A summary of Tables 3 and 5 is in Tables 4 and 6, respectively. In particular here, it is synthesized the comparison between RedistProx and the best between the two implementations of NCVX$^{\text{penalty}}$. Generally speaking, RedistProx appears to work well in terms of number of function evaluations, whereas NCVX$^{\text{penalty}}$ offers a quite reliable behaviour both in terms of number of function evaluations and precision.

Funding

This work has been partially supported by RFBR [grant number 13-07-12010].

References

[1] Astorino A, Fuduli A. Nonsmooth optimization techniques for semisupervised classification. IEEE Trans. Pattern Anal. Machine Intel. 2007;29:2135–2142.
[2] Astorino A, Fuduli A, Gaudioso M. DC models for spherical separation. J. Global Optimiz. 2010;48:657–669.
[3] Astorino A, Fuduli A, Gaudioso M. Margin maximization in spherical separation. Comput. Optimiz. Appl. 2012;53:301–322.
[4] Astorino A. Conic separation of finite sets. I. The homogeneous case. J. Convex Anal., to appear.
[5] Bergeron C, Moore G, Zaretzki J, Breneman C, Bennett K. Fast bundle algorithm for multiple instance learning. IEEE Trans. Pattern Anal. Machine Intel. 2012;34:1068–1079.
[6] Le Thi HA, Le HM, Pham Dinh T, Van Huynh N. Binary classification via spherical separator by DC programming and DCA. J. Global Optimiz. 2013;56: 1393–1407.
[7] Astorino A, Fuduli A, Gorgone E. Non-smoothness in classification problems. Optimiz. Methods Softw. 2008;23:675–688.
[8] Bagirov AM, Rubinov AM, Soukhoroukova NV, Yearwood J. Unsupervised and supervised data classification via nonsmooth and global optimization. TOP. 2003;11:1–75.
[9] Karmitsa N, Bagirov AM, Mäkelä M. Comparing different nonsmooth minimization methods and software. Optimiz. Methods Softw. 2012;27:131–153.
[10] Astorino A, Frangioni A, Fuduli A, Gorgone E. A nonmonotone proximal bundle method with (potentially) continuous step decisions. SIAM J. Optimiz. 2013;23:1784–1809.
[11] Astorino A, Frangioni A, Gaudioso M, Gorgone E. Piecewise-quadratic approximations in convex numerical optimization. SIAM J. Optimiz. 2011;21:1418–1438.
[12] Daniilidis A, Sagastizábal C, Solodov M. Identifying structure of nonsmooth convex functions by the bundle technique. SIAM J. Optimiz. 2009;20:820–840.
[13] Demyanov AV, Fuduli A, Miglionico G. A bundle modification strategy for convex minimization. Eur. J. Oper. Res. 2007;180:38–47.
[14] Fuduli A, Gaudioso M. Tuning strategy for the proximity parameter in convex minimization. J. Optimiz. Theory Appl. 2006;130:95–112.
[15] Hou L, Sun W. On the global convergence of a nonmonotone proximal bundle method for convex nonsmooth minimization. Optimiz. Methods Softw. 2008;23:227–235.

[16] Karas E, Ribeiro A, Sagastizábal C, Solodov M. A bundle-filter method for nonsmooth convex constrained optimization. Math. Program. 2009;116:297–320.
[17] Nurminski EA. Separating plane algorithms for convex optimization. Math. Program. B. 1997;76:373–391.
[18] Sagastizábal C, Solodov M. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. SIAM J. Optimiz. 2006;16:146–169.
[19] Solodov MV. A bundle method for a class of bilevel nonsmooth convex minimization problems. SIAM J. Optimiz. 2007;18:242–259.
[20] Gaudioso M, Giallombardo G, Miglionico G. An incremental method for solving convex finite min–max problems. Math. Oper. Res. 2006;31:173–187.
[21] Gaudioso M, Giallombardo G, Miglionico G. On solving the lagrangian dual of integer programs via an incremental approach. Comput. Optimiz. Appl. 2009;44:117–138.
[22] Kiwiel KC, Lemaréchal C. An inexact bundle variant suited to column generation. Math. Program. 2009;118:177–206.
[23] Nedic A, Bertsekas DP. The effect of deterministic noise in subgradient methods. Math. Program. 2010;125:75–99.
[24] Frangioni A. Generalized bundle methods. SIAM J. Optimiz. 2003;13:117–156.
[25] Mäkelä M. Survey of bundle methods for nonsmooth optimization. Optimiz. Methods Softw. 2002;17:1–29.
[26] Hiriart-Urruty JB, CLemaréchal 1993. *Convex analysis and minimization algorithms. Vols. I–II*. Berlin: Springer-Verlag.
[27] Haarala N, Miettinen K, Mäkelä MM. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. Math. Program. 2007;109:181–205.
[28] Hare W, Sagastizábal C. Computing proximal points of nonconvex functions. Math. Program. 2009;116:221–258.
[29] Hare W, Sagastizábal C. A redistributed proximal bundle method for nonconvex optimization. SIAM J. Optimiz. 2010;10:2442–2473.
[30] Karmitsa N, Mäkelä M. Adaptive limited memory bundle method for bound constrained large-scale nonsmooth optimization. Optimization. 2010;59:945–962.
[31] Karmitsa N, Mäkelä M. Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. Optimiz. Methods Softw. 2010;25:895–916.
[32] Noll D. Cutting plane oracles to minimize non-smooth non-convex functions. Set-Valued Variat. Anal. 2010;18:531–568.
[33] Schramm H, Zowe J. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis. SIAM J. Optimiz. 1992;1:121–152.
[34] Lewis AS, Overton ML. Nonsmooth optimization via quasi-Newton methods. Math. Program. 2013;141:135–163.
[35] Luksăn L, Vlček J. A bundle-Newton method for nonsmooth unconstrained minimization. Math. Program. 1998;83:373–391.
[36] Burke JB, Lewis AS, Overton ML. Approximating subdifferentials by random sampling of gradients. Math. Oper. Res. 2002;27:567–584.
[37] Burke JV, Lewis AS, Overton ML. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optimiz. 2005;15:751–779.
[38] Kiwiel KC. Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optimiz. 2007;18:379–388.
[39] Kiwiel KC. A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optimiz. 2010;20:1983–1994.
[40] Bagirov AM, Karasozen B, Sezer M. Discrete gradient method: derivative-free method for nonsmooth optimization. J. Optimiz. Theory Appl. 2008;137:317–334.
[41] Karmitsa N. Limited memory discrete gradient bundle method for nonsmooth derivative-free optimization. Optimization. 2012;61:1491–1509.

[42] Apkarian P, Noll D, Prot O. A trust region spectral bundle method for nonconvex eigenvalue optimization. SIAM J. Optimiz. 2008;19:281–306.
[43] Apkarian P, Noll D, Prot O. A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. J. Convex Anal. 2009;16:641–666.
[44] Fuduli A, Gaudioso M, Giallombardo G. A DC piecewise affine model and bundling technique in nonconvex nonsmooth minimization. Optimiz. Methods Softw. 2004;19:89–102.
[45] Fuduli A, Gaudioso M, Giallombardo G. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM J. Optimiz. 2004;14:743–756.
[46] Gaudioso M, Gorgone E, Monaco MF. Piecewise linear approximations in nonconvex nonsmooth optimization. Numer. Math. 2009;113:73–88.
[47] Cheney EW, Goldstein AA. Newton's method for convex programming and Tchebycheff approximation. Numer. Math. 1959;1:253–268.
[48] Kelley JE. The cutting-plane method for solving convex programs. J. SIAM. 1960;8:703–712.
[49] Frangioni A. Solving semidefinite quadratic problems within nonsmooth optimization algorithms. Comput. Oper. Res. 1996;23:1099–1118.
[50] Kiwiel KC. A method for solving certain quadratic programming problems arising in nonsmooth optimization. IMA J. Numer. Anal. 1986;6:137–152.
[51] Nurminski EA. Convergence of the suitable affine subspace method for finding the least distance to a simplex. Comput. Math. Math. Phys. 2005;45:1915–1922.
[52] Nurminski EA. Projection onto polyhedra in outer representation. Comput. Math. Math. Phys. 2008;48:367–375.
[53] Kiwiel KC. An aggregate subgradient method for nonsmooth convex minimization. Math. Program. 1983;27:320–341.
[54] Lukšan L 2000. Test problems for nonsmooth unconstrained and linearly constrained optimization. Tech. Rep. 798. Prague: Institute of Computer Science, Academy of Sciences of the Czech Republic.
[55] Ferrier C. Bornes duales de problémes d'optimisation polynomiaux [PhD thesis]. Toulose, France: Laboratoire Approximation et Optimisation, Université Paul Sabatier; 1997.