SIAM J. Optimization
Vol. xx, No. x, pp. xx–xx, 200x

AN INFEASIBLE BUNDLE METHOD FOR NONSMOOTH CONVEX CONSTRAINED OPTIMIZATION WITHOUT A PENALTY FUNCTION OR A FILTER [*]

CLAUDIA SAGASTIZÁBAL [†] AND MIKHAIL SOLODOV [‡]

Abstract. Global convergence in constrained optimization algorithms has traditionally been enforced by the use of parametrized penalty functions. Recently, the filter strategy has been introduced as an alternative. At least part of the motivation for filter methods consists in avoiding the need for estimating a suitable penalty parameter, which is often a delicate task. In this paper, we demonstrate that the use of a parametrized penalty function in nonsmooth convex optimization can be avoided without using the relatively complex filter methods. We propose an approach which appears to be more direct and easier to implement, in the sense that it is closer in spirit and structure to the well-developed unconstrained bundle methods. Preliminary computational results are also reported.

Key words. constrained optimization, nonsmooth convex optimization, bundle methods.

AMS subject classifications. 90C30, 65K05, 49D27

1. Introduction and motivation.

We consider the problem

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} f(x) \\ c(x) \le 0\,, \end{cases} \tag{1.1}$$

where $f, c : \mathbb{R}^n \to \mathbb{R}$ are convex functions, in general nondifferentiable. We note that there is no loss of generality in formulating (1.1) with only one constraint: if necessary, c can be defined as the pointwise maximum of finitely many convex functions, thus covering the case of multiple inequality constraints. In our development, we assume that the Slater constraint qualification [27] holds, i.e., there exists $x \in \mathbb{R}^n$ such that $c(x) < 0$. We also assume that an *oracle* is available, which for any given $x \in \mathbb{R}^n$ computes the values $f(x)$ and $c(x)$, and *one* subgradient for each of the functions, i.e., some $g_f \in \partial f(x)$ and some $g_c \in \partial c(x)$. We do not assume that there is any control over which particular subgradients are computed by the oracle (for example, for problems with more than one constraint, i.e., when c is defined by the maximum operation, we may have subgradient information about only one constraint among those with the largest value).

Nonsmooth optimization (NSO) problems are in general difficult to solve, even when they are unconstrained. Among algorithms for NSO, we mention the subgradient, cutting-planes, analytic center cutting-planes (ACCP) and bundle methods (see [36], [5, 15], [11] and [13, 35], respectively). Bundle and ACCP methods are stabilized versions of the cutting-planes method, and they are currently recognized as the most robust and reliable NSO algorithms. ACCP methods are based on information given by a certain separation procedure, which puts it somewhat outside of the oracle

[*] Received by the editors February, 2004; revised November 2004.

[†] Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil (sagastiz@impa.br) On leave from INRIA-Rocqcencourt, BP 105, 78153 Le Chesnay, France. Research of this author was supported by CNPq under Grant 303540/03-6, by PRONEX-Optimization, and by FAPERJ.

[‡] Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil (solodov@impa.br). Research of this author is supported in part by CNPq Grants 300734/95-6(RN) and 471780/2003-0, by PRONEX-Optimization, and by FAPERJ.

framework considered here. In this paper we focus on bundle methods, specifically on their proximal form.

For unconstrained problems, iterates of a proximal bundle algorithm are generated by solving a quadratic programming problem (QP). Each QP is defined by means of a cutting-planes model of the objective function, stabilized by a quadratic term centered at the best point obtained so far (which is referred to as the last *descent* or *serious step*). An important feature of bundle methods is that the size of each QP can be controlled via the so-called *aggregation* techniques, see for instance [2, Ch. 9] and also Section 2 below. We emphasize that the latter is crucial for any practical implementation.

Constrained nonsmooth problems are more complex, and only a few practical methods can be found in the literature. Convex problems with "easy" constraints (such as bound or linear constraints) can be solved either by inserting the constraints directly into each QP, or by projecting iterates onto the feasible set, see for instance [10] and [19, 20]. For general convex constrained problems, like problem (1.1) considered here, one possibility is to solve an equivalent unconstrained problem with an exact penalty objective function, see [17, 22]. This approach, however, presents some drawbacks which are typical whenever a penalty function is employed. Specifically, estimating a suitable value of the penalty parameter is sometimes a delicate task. Furthermore, if a large value of the parameter is required to guarantee the exactness of a given penalty function, then numerical difficulties arise.

More recently, [7] proposed the filter strategy [8] as an alternative to the use of a penalty function in the framework of bundle methods for solving (1.1). However, the development of [7] is quite involved, and in particular, the resulting method appears considerably more complicated when compared, for example, to standard bundle methods for the unconstrained case. Furthermore, techniques for bundle compression and aggregation, although mentioned in [7], are not explicitly addressed. As stated, the method of [7] does not guarantee that the number of constraints in the subproblems can be kept smaller than a given desired bound, even if "inactive cuts" are removed from the bundle. Without this feature, a method cannot be guaranteed to be practical.

As other bundle-type methods for (1.1) that do not use penalization, we mention [28, 29] and [16, Ch. 5]. But it should be emphasized that in the cited methods all the serious iterates are required to be feasible, including the starting point. Therefore, there are no concerns associated with the use of penalty functions, neither the need for alternative strategies, such as filter methods. It should be noted that feasible methods suffer from an important drawback: computing a feasible point is required to start the algorithm. This "phase I" general (nonsmooth) convex feasibility problem may be as difficult to solve as (1.1) itself. As a result, the overall computational burden of solving the problem may increase considerably. On the other hand, feasible methods can be useful in applications where problem function(s) may not be defined everywhere outside of the feasible region. We point out that our method, if started from a feasible point, stays feasible (see Proposition 4.1) and thus can operate "in feasible mode", if an appropriate starting point is provided.

Before proceeding with our discussion, we introduce the *improvement function* associated with problem (1.1). For a given $x \in \mathbb{R}^n$, let

$$h_x(y) := \max\{f(y) - f(x),\, c(y)\}, \quad y \in \mathbb{R}^n. \tag{1.2}$$

Among other things, it holds that $\bar{x}$ is a solution to (1.1) if, and only if, $\bar{x}$ solves the unconstrained problem of minimizing $h_{\bar{x}}$ (see Theorem 2.1 below). The use of $h_{\bar{x}}$ as a theoretical tool in the convergence analysis of bundle-type methods can be traced back to [28], see also [16]. However, in none of these works the improvement function is used in the algorithms themselves. In addition, since in [28, 16] the infeasible iterates are automatically declared "null steps", the test to accept an iterate as the next serious step involves the objective function f only. Thus the resulting sequence of serious steps is both feasible and monotone in f. The piecewise linearization of the improvement function has also been used in the methods of feasible directions for solving *smooth* problems (e.g., [38, 32]). However, those methods are again feasible and the improvement function itself is not involved in the algorithms.

Infeasible bundle methods are very rare. Prior to [7], we could find in the literature only the "phase I-phase II" modification of the feasible method in [16, Ch. 5.7] and the constrained level bundle methods of [24]. In [24], successive approximations of the exact improvement function $h_{\bar{x}}$ are used in the algorithm. Specifically, in the expression

$$h_{\bar{x}}(y) = \max\{f(y) - f(\bar{x}),\, c(y)\} = \lambda f(y) + (1-\lambda)c(y) - \lambda f(\bar{x}) \text{ for some } \lambda \in [0,1],$$

the values of λ and of $f(\bar{x})$ are estimated at each iteration. Those estimates are used to define a certain gap function and an associated level parameter for the QP. It is well-known that level methods are especially suitable for those problems where the optimal value $f(\bar{x})$ is either known or is easy to estimate. This certainly is not true in general. In fact, estimating the optimal value is a delicate issue and inappropriately chosen values may lead to infeasible QPs.

In this paper, we propose an infeasible proximal bundle method for solving (1.1), which does not use either a penalty function or a filter. With respect to [29, 16], the advantage is that computing a feasible point is not needed to start the algorithm. Also, since serious steps can be infeasible, monotonicity in f is not enforced (outside of the feasible set). Rather, there is a balance between the search for feasibility and for the reduction of the objective function. But this balance is followed in a manner different from the filter strategy. We also emphasize that compared to [7], our method is much closer to the well-developed unconstrained bundle methods, and thus is easier to implement. For example, we can manage the size of QPs by a suitable modification of the standard aggregation techniques. Finally, compared to [24], QPs in our method are always feasible, independently of the choice of parameters.

Our approach can be viewed as an unconstrained proximal bundle method [13, 16, 2] applied to the function $h_x(\cdot)$ directly, with the important distinction that x is the last serious step, and thus, the function being minimized varies along the iterations, see Section 3 for details. We emphasize that serious steps need neither be monotone in f nor feasible. Of course, the fact that the improvement function changes along the iterations makes standard convergence results not applicable directly, and specific analysis is needed. Actually, some subtle modifications are needed also in the bundle method itself. Nevertheless, our approach is quite close to standard unconstrained bundle methods. Apart from leading to a relative ease in the computer implementation, this also opens the potential of extending various results obtained for the unconstrained bundle methods to the constrained case, e.g., the variable metric [1, 25, 26] and quasi-Newton [31, 4] techniques, methods with inexact data [12, 37], etc.

The paper is organized as follows. In Section 2, we state some basic properties of the improvement function, and also give an overview of proximal bundle methods for the unconstrained case, including the aggregation and compression technique. This is done in order to set the notation for the algorithm, and also to build a link from the well-known unconstrained method to the constrained one. The algorithm itself is stated in Section 3, where also some preliminary properties are established. Convergence analysis is provided in Section 4, and computational experience is reported in Section 5.

Our notation is fairly standard. The Euclidean inner product in $\mathbb{R}^n$ is denoted by $\langle x, y\rangle = \sum_{j=1}^n x_j y_j$, and the associated norm by $\|\cdot\|$. The positive-part function is denoted by $x^+ := \max\{x, 0\}$. For a set X in $\mathbb{R}^n$, $\operatorname{conv} X$ denotes its convex hull. By $\partial_\varepsilon h(x)$ we denote the ε-subdifferential of a convex function h at the point $x \in \mathbb{R}^n$, with $\partial_0 h(x) = \partial h(x)$ being the usual subdifferential.

2. Preliminaries.

We start with the properties of the improvement function to be used in the sequel. Next, we discuss some basics of the standard bundle methods, mainly to fix notation and remind the reader the principal relations. Also, we use this discussion to point out where appropriate modifications would be needed when passing from the unconstrained to the constrained case. No proofs are given in this section. Proofs and calculations are worked out in detail for the constrained algorithm in Section 4.

2.1. The improvement function.

Directly by the definition (1.2), the subdifferential of the improvement function is given by

$$\partial h_x(y) = \begin{cases} \partial f(y) & \text{if } f(y) - f(x) > c(y)\,, \\ \operatorname{conv}\{\partial f(y) \bigcup \partial c(y)\} & \text{if } f(y) - f(x) = c(y)\,, \\ \partial c(y) & \text{if } f(y) - f(x) < c(y)\,. \end{cases} \qquad (2.1)$$

In addition, we have that

$$h_x(x) = c^+(x) = \max\{c(x), 0\} \quad \text{for all } x \in \mathbb{R}^n.$$

Finally (e.g., [16, Lemma 2.16, p.17]), the following holds.

THEOREM 2.1. *Suppose that the Slater constraint qualification is satisfied for* (1.1). *Then the following statements are equivalent:*

(i) $\bar{x}$ *is a solution to* (1.1)*;*

(ii) $\min\{h_{\bar{x}}(y) \mid y \in \mathbb{R}^n\} = h_{\bar{x}}(\bar{x}) = 0$*;*

(iii) $0 \in \partial h_{\bar{x}}(\bar{x})$*, i.e.,* $0 \in \partial\varphi(\bar{x})$*, where* $\varphi(\cdot) := h_{\bar{x}}(\cdot)$*.*

2.2. An overview of unconstrained bundle methods.

Consider, for the moment, the unconstrained problem

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

where $h(\cdot)$ is some fixed convex function. For the sake of simplicity, we also suppose for now that there is no bundle compression/aggregation in the algorithm. We refer the reader to [2, Ch. 9.3] for proofs of the relations stated in this subsection.

Let ℓ be the current iteration index. Bundle methods keep memory of the past in a *bundle* of information

$$\mathcal{B}_\ell := \bigcup_{i<\ell} \left\{ \left(y^i, h_i = h(y^i)\,, g_h^i \in \partial h(y^i)\right) \right\} \text{ and } (x^k, h(x^k)),\ k = k(\ell),$$

where $k(\ell)$ denotes the index of the last "serious" step preceding the iteration ℓ. Serious iterates, also called stability centers, form a subsequence $\{x^k\} \subset \{y^i\}$ such that $\{h(x^k)\}$ is strictly decreasing. This will be made more precise later.

We mention two peculiarities of our notation. When it is clear from the context, we shall not be explicitly specifying the dependence of k on the current iteration index ℓ. Also, in the sequel we shall write $i \in \mathcal{B}_\ell$ to mean that there exists an element in the set $\mathcal{B}_\ell$ indexed by i. Although this notation is formally improper, it does not lead to any confusion, while simplifying some relations below.

The bundle of past information is used to define at each iteration a cutting-planes model of the objective function,

$$\psi_\ell(y) := \max_{i\in\mathcal{B}_\ell} \left\{ h_i + \left\langle g_h^i, y - y^i \right\rangle \right\} .$$

An equivalent expression, better suited for implementations, centers the cutting-planes model at the stability center x^k:

$$\psi_\ell(y) = h(x^k) + \max_{i\in\mathcal{B}_\ell} \left\{ -e_i^k + \left\langle g_h^i, y - x^k \right\rangle \right\} , \tag{2.2}$$

where the terms e_i^k are the (nonnegative) linearization errors

$$e_i^k := h(x^k) - h_i - \left\langle g_h^i, x^k - y^i \right\rangle .$$

In particular,

$$g_h^i \in \partial_{e_i^k} h(x^k),$$

i.e., $h(y) \geq h(x^k) + \left\langle g_h^i, y - x^k \right\rangle - e_i^k$ for all $y \in \mathbb{R}^n$.

Since the linearization errors depend on x^k, they need to be updated every time x^k changes (for this reason, they are indexed by both k and i). For further reference, note that the linearization errors obviously "depend" also on h (h is fixed in this section, but not in the rest of the paper). Thus in the update of the linearization errors in our algorithm, we shall also have to account for an eventual change in h.

The advantage of expressing the model in the form of (2.2) is that it requires less memory for storing the relevant information: the bundle becomes

$$\mathcal{B}_\ell = \bigcup_{i<\ell} \left\{ \left(e_i^k \in \mathbb{R}_+, g_h^i \in \partial_{e_i^k} h(x^k) \right) \right\} \text{ and } (x^k, h(x^k)).$$

Given μ_ℓ, a positive proximal parameter, the next iterate y^ℓ is generated by solving a QP reformulation of the problem

$$\min_{y\in\mathbb{R}^n} \psi_\ell(y) + \frac{1}{2}\mu_\ell \|y - x^k\|^2.$$

Clearly, y^ℓ is unique. Furthermore, it is characterized by the following conditions (see [2, Lemma 9.8]):

$$y^\ell = x^k - \frac{1}{\mu_\ell}\hat{g}^\ell, \text{ where } \hat{g}^\ell \in \partial\psi_\ell(y^\ell),$$

$$\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h(x^k), \text{ where } \hat{\varepsilon}_\ell^k = h(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell}\|\hat{g}^\ell\|^2 \geq 0.$$

An iterate y^ℓ is considered good enough to become the next serious step when $h(y^\ell)$ provides a significant decrease (with respect to $h(x^k)$), measured in terms of a nominal decrease. Specifically, let $m \in (0,1)$ be a given parameter. The nominal decrease is defined by

$$\delta_\ell := h(x^k) - \psi_\ell(y^\ell) - \frac{1}{2}\mu_\ell \|y^\ell - x^k\|^2 = \hat\varepsilon_\ell^k + \frac{1}{2\mu_\ell}\|\hat g^\ell\|^2 \geq 0.$$

When y^ℓ satisfies the descent test

$$h(y^\ell) \leq h(x^k) - m\delta_\ell, \tag{2.3}$$

a serious step is declared: $x^{k+1} = y^\ell$. Otherwise, y^ℓ is declared a null step and x^k remains unchanged.

The algorithm stops when δ_ℓ is small enough when compared to a given tolerance. In this case, both $\hat\varepsilon_\ell^k$ and $\|\hat g^\ell\|$ are small and, since $\hat g^\ell \in \partial_{\hat\varepsilon_\ell^k} h(x^k)$, for any $M > 0$ and all $y \in \mathbb{R}^n$ such that $\|y - x^k\| \leq M$, the approximate optimality condition $h(y) \geq h(x^k) - \hat\varepsilon_\ell^k - M\|\hat g^\ell\|$ holds.

We next consider the effect of compressing the bundle.

2.3. Aggregation technique. The number of constraints in the QP used to generate y^ℓ is precisely the number of elements in the bundle $\mathcal{B}_\ell$. Obviously, one has to keep this number computationally manageable. Thus, the bundle has to be *compressed* when the number of elements reaches some chosen bound. This has to be done without impairing convergence of the algorithm. For this purpose, the so-called *aggregate* function is fundamental:

$$l_{k,\ell}(y) := h(x^k) - \hat\varepsilon_\ell^k + \langle \hat g^\ell, y - x^k\rangle, \quad k = k(\ell).$$

Note that this function can be defined directly from the aggregate couple $(\hat\varepsilon_\ell^k, \hat g^\ell \in \partial_{\hat\varepsilon_\ell^k} h(x^k))$. Alternatively, the same information can be retrieved from all the "active" bundle elements, i.e., those defining $\psi_\ell(y^\ell)$.

Before looping from ℓ to $\ell+1$, the next bundle $\mathcal{B}_{\ell+1}$ is defined. If the bundle has reached its maximum allowed size, it must be compressed. Reducing the bundle amounts to replacing (at iteration $\ell+1$) the cutting-planes model (2.2) by another function, defined with a smaller number of cutting planes, which we shall still denote by $\psi_{\ell+1}$. As pointed out in [6, Section 4, eqs. (4.7)-(4.9)], one can use any collection of functions satisfying (for all $y \in \mathbb{R}^n$) the following three conditions:

$$\begin{aligned}
\psi_\ell(y) &\leq h(y) && \text{for all } \ell \geq 1\,, && \text{(2.4a)}\\
l_{k(\ell),\ell}(y) &\leq \psi_{\ell+1}(y) && \text{for those } \ell \text{ for which } y^\ell \text{ is a null step,} && \text{(2.4b)}\\
h_\ell + \langle g_h^\ell, y - y^\ell\rangle &\leq \psi_{\ell+1}(y) && \text{for those } \ell \text{ for which } y^\ell \text{ is a null step.} && \text{(2.4c)}
\end{aligned}$$

We note that (2.4a) will not be automatic in our setting. Indeed, as already commented, the function h will be changing after every serious step. As a consequence, (2.4a) can be violated unless appropriate care is taken.

Suppose, however, that (2.4a) holds. In terms of bundle information, the remaining conditions mean that it is enough for the new bundle to contain both the aggregate information (to ensure (2.4b)) and the last generated information (to ensure (2.4c)). These values are, respectively, $(\hat\varepsilon_\ell^k, \hat g^\ell)$ and $(y^\ell, h_\ell, g_h^\ell \in \partial h(y^\ell))$. In particular, at any iteration, the bundle can contain as few elements as we wish (as long as the two specified above are included). Note also that if the bundle is not to be compressed at the current iteration, then the aggregate information is redundant (because it is already contained in the bundle elements which are active in the QP subproblem).

Accordingly, we shall write the next bundle in the form

$$\mathcal{B}_{\ell+1} := \mathcal{B}^{oracle}_{\ell+1} \bigcup \mathcal{B}^{agg}_{\ell+1} \quad \text{and } (x^k, h(x^k)),\ k = k(\ell+1), \text{ the last “serious” iterate},$$

where the “oracle” bundle is any set such that

$$\Big\{ \big(e^k_\ell, g^\ell_h\big) \Big\} \subseteq \mathcal{B}^{oracle}_{\ell+1} \subseteq \bigcup_{i\le\ell} \Big\{ \Big(e^k_i \in \mathbb{R}_+, g^i_h \in \partial_{e^k_i} h(x^k)\Big) \Big\},$$

while the “aggregate” bundle satisfies

$$\Big\{ \big(\hat\varepsilon^k_\ell, \hat g^\ell\big) \Big\} \subseteq \mathcal{B}^{agg}_{\ell+1} \subseteq \bigcup_{i\le\ell} \Big\{ \Big(\hat\varepsilon^k_i \in \mathbb{R}_+, \hat g^i \in \partial_{\hat\varepsilon^k_i} h(x^k)\Big) \Big\}.$$

The left-most inclusions in the last two relations above need to be specified explicitly only when there is bundle compression at the ℓ-th iteration (if there is no compression, they hold automatically, because of the right-most inclusions). We note that similarly to updating the linearization errors e^k_i, the quantities $\hat\varepsilon^k_i$ also need to be updated every time when k changes, see (2.5) and (3.6) below.

The next cutting-planes model is then defined by

$$\psi_{\ell+1}(y) = h(x^k) + \max\Bigg\{ \max_{i\in\mathcal{B}^{oracle}_{\ell+1}} \{-e^k_i + \langle g^i_h, y - x^k\rangle\}\ ,\ \max_{i\in\mathcal{B}^{agg}_{\ell+1}} \{-\hat\varepsilon^k_i + \langle \hat g^i, y - x^k\rangle\} \Bigg\},$$
$$k = k(\ell+1).$$

As already mentioned, every time a new serious step has been declared, both linearization and aggregate errors need to be modified. The update aims at satisfying the key relations

$$g^i_h \in \partial_{e^k_i} h(x^{k+1}) \quad \text{and} \quad \hat g^i \in \partial_{\hat\varepsilon^k_i} h(x^{k+1}),$$

which should hold for all elements in the new bundle. The following simple updating formulæ guarantee the required properties (when h is fixed):

$$\left\{ \begin{array}{ll} e^{k+1}_i := e^k_i + h(x^{k+1}) - h(x^k) + \langle g^i_h, x^k - x^{k+1}\rangle & \text{if } i \in \mathcal{B}^{oracle}_{\ell+1}, \\ \hat\varepsilon^{k+1}_i := \hat\varepsilon^k_i + h(x^{k+1}) - h(x^k) + \langle \hat g^i, x^k - x^{k+1}\rangle & \text{if } i \in \mathcal{B}^{agg}_{\ell+1}\,. \end{array} \right. \tag{2.5}$$

We next show how to adapt the unconstrained bundle methodology described above to solving the constrained problem (1.1).

3. Defining the Constrained Algorithm.

Given the last serious iterate x^k (we note that the starting point x^0 is considered a serious iterate), we apply an unconstrained proximal bundle method to the function $h(\cdot) := h_k(\cdot) = h_{x^k}(\cdot)$, until the next serious iterate x^{k+1} is generated. At this time, we change $h(\cdot)$ to $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$, make the necessary changes to the bundle, and repeat the process. We point out that the development is not straightforward. For one thing, it is possible that $f(x^{k+1}) > f(x^k)$. As is easy to observe, in that case we have $h_{k+1}(\cdot) \le h_k(\cdot)$. As a consequence, the accumulated cutting-planes model for $h_k(\cdot)$ may not be a valid (lower) approximation for $h_{k+1}(\cdot)$. Thus, the model has to be revised and adjusted to ensure that conditions (2.4a)-(2.4c) (in particular (2.4a)) are satisfied for the new $h(\cdot) := h_{k+1}(\cdot)$. Note that this adjustment is independent of compressing the bundle, which will require additional care.

In the following, we explain how to build the model ψ_ℓ satisfying (2.4a)-(2.4c), even when $h(\cdot)$ changes at a serious step.

3.1. Bundle information. Since $h(\cdot)$ varies with k, past information relevant for constructing the model is no longer just (e_i, g_h^i). In particular, separate information about the objective and constraint functions needs to be kept. This information is $(f_i = f(y^i), c_i = c(y^i))$ and $(g_f^i \in \partial f(y^i), g_c^i \in \partial c(y^i))$. Or, equivalently, $(e_{f_i}^k, e_{ci}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k), g_c^i \in \partial_{e_{ci}^k} c(x^k))$, where the linearization errors for f and c, respectively, are

$$\begin{array}{rcl} e_{f_i}^k & := & f(x^k) - f_i - \langle g_f^i, x^k - y^i\rangle, \\ e_{ci}^k & := & c(x^k) - c_i - \langle g_c^i, x^k - y^i\rangle. \end{array} \tag{3.1}$$

The purpose of keeping the bundle information separated is twofold:

– First, knowing (f_i, c_i) makes it possible to compute the function and subgradient values for different functions h, see Lemma 3.1 below.

– Second, as shown in Lemma 3.2 below, separate linearization errors can be updated by a simple formula, even when h changes.

Therefore, we define

$$\begin{array}{l} \mathcal{B}_\ell := \mathcal{B}_\ell^{oracle} \bigcup \mathcal{B}_\ell^{agg} \quad \text{and} \quad (x^k, f(x^k), c(x^k)),\ k = k(\ell), \text{ the last “serious” iterate,} \\ \text{with } \mathcal{B}_\ell^{oracle} \subseteq \displaystyle\bigcup_{i<\ell} \left\{ \left(f_i, c_i, e_{f_i}^k, e_{ci}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k), g_c^i \in \partial_{e_{ci}^k} c(x^k) \right) \right\} \\ \text{and } \mathcal{B}_\ell^{agg} \subseteq \displaystyle\bigcup_{i<\ell} \left\{ \left(\hat\varepsilon_i^k, \hat g^i \in \partial_{\hat\varepsilon_i^k} h_k(x^k) \right) \right\}. \end{array} \tag{3.2}$$

LEMMA 3.1. *In the notation of* (3.1) *and* (3.2)*, for each* $i \in \mathcal{B}_\ell^{oracle}$*, define*

$$\left\{ \begin{array}{llll} e_i^k := e_{f_i}^k + c^+(x^k) & \textit{and} \quad g_{h_k}^i := g_f^i\,, & & \textit{if } f_i - f(x^k) \ge c_i, \\ e_i^k := e_{ci}^k + c^+(x^k) - c(x^k) & \textit{and} \quad g_{h_k}^i := g_c^i\,, & & \textit{if } f_i - f(x^k) < c_i. \end{array} \right. \tag{3.3}$$

Then $e_i^k \ge 0$ *and* $g_{h_k}^i \in \partial_{e_i^k} h_k(x^k)$.

Proof. By (3.1) and the convexity of f and c, $e_{f_i}^k \ge 0$ and $e_{ci}^k \ge 0$. Since also $c^+(x^k) \ge 0$ and $c^+(x^k) - c(x^k) \ge 0$, (3.3) implies that $e_i^k \ge 0$.

Recalling that $h_k(x^k) = c^+(x^k)$, we have to show that for all $y \in \mathbb{R}^n$, it holds that $h_k(y) \ge c^+(x^k) + \langle g_{h_k}^i, y - x^k\rangle - e_i^k$. By using the definitions of h_k, of the subdifferential, and of the errors $e_{f_i}^k$, e_{ci}^k, we obtain that

$$\begin{array}{rcl} h_k(y) & = & \max \left\{ \begin{array}{l} f(y) - f(x^k) \\ c(y) \end{array} \right. \\ & \ge & \max \left\{ \begin{array}{l} f_i - f(x^k) + \langle g_f^i, y - y^i\rangle \\ c_i + \langle g_c^i, y - y^i\rangle \end{array} \right. \\ & = & \max \left\{ \begin{array}{l} \langle g_f^i, y - x^k\rangle - e_{f_i}^k \\ c(x^k) + \langle g_c^i, y - x^k\rangle - e_{ci}^k. \end{array} \right. \end{array}$$

By adding and subtracting $c^+(x^k)$ in the right-hand side of the relation above, and using the definition of $g_{h_k}^i$, we obtain that

$$h_k(y) \ge c^+(x^k) + \langle g_{h_k}^i, y - x^k\rangle - \left\{ \begin{array}{ll} \left(e_{f_i}^k + c^+(x^k)\right), & \text{if } f_i - f(x^k) \ge c_i, \\ \left(e_{ci}^k + c^+(x^k) - c(x^k)\right), & \text{if } f_i - f(x^k) < c_i. \end{array} \right.$$

The result now follows from the definition of e_i^k in (3.3). □

The cutting-planes model associated with (3.2), (3.3) is given by

$$\psi_\ell(y) = c^+(x^k) + \max\left\{ \max_{i\in\mathcal{B}_\ell^{oracle}} \{-e_i^k + \langle g_{h_k}^i, y - x^k\rangle\} \ , \max_{i\in\mathcal{B}_\ell^{agg}} \{-\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k\rangle\} \right\},$$
$$k = k(\ell). \tag{3.4}$$

For this model to satisfy (2.4a)-(2.4c) when passing to the iteration $\ell+1$, we consider separately the two cases of the ℓ-th iteration being a null step or a serious step.

Suppose first that the QP subproblem defined with ψ_ℓ given by (3.4) generates y^ℓ as a null step. By construction, the new bundle satisfies (3.2) and (3.3) written with ℓ replaced by $\ell+1$ (k remains the same). Thus, Lemma 3.1 holds, and $g_{h_k}^i \in \partial_{e_i^k} h_k(x^k)$ for all $i \in \mathcal{B}_{\ell+1}^{oracle}$. Likewise, aggregate subgradients satisfy the inclusion $\hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h_k(x^k)$ for all $i \in \mathcal{B}_{\ell+1}^{agg}$. Therefore, (2.4a) (written with ℓ replaced by $\ell+1$) is automatically satisfied. Finally, for conditions (2.4b) and (2.4c) to hold, it is enough to make sure that

$$\left\{\left(e_\ell^k, g_{h_k}^\ell \in \partial_{e_\ell^k} h_k(x^k)\right)\right\} \subseteq \mathcal{B}_{\ell+1}^{oracle} \text{ and}$$

$$\left\{\left(\hat{\varepsilon}_\ell^k, \hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k)\right)\right\} \subseteq \mathcal{B}_{\ell+1}^{agg} \text{ , if there is compression.}$$

Those inclusions are also automatically satisfied, if the bundle is managed as in any standard method; see Step 4 in Algorithm 3.1 below.

Therefore, when there is a null step, the update of the bundle (and of the model) does not present any problem. This is as expected, since the function $h(\cdot) = h_k(\cdot)$ is fixed between consecutive serious steps. The situation changes when y^ℓ is declared a serious step. Specifically, the aggregate bundle elements need a special update. We next discuss this case.

3.2. Adjusting the Model After a Serious Step. Suppose that for some iteration ℓ the descent test is satisfied (i.e., condition (2.3) written with h replaced by h_k), so that a new stability center $x^{k+1} = y^\ell$ is generated. This means, in particular, that at the next iterate we start working with the new function $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$.

As mentioned in [6], conditions (2.4a)-(2.4c) guarantee that the bundle technique applied to the new function $h(\cdot) = h_{k+1}(\cdot)$ either produces a descent step after a finite number of null steps, or the point x^{k+1} is a minimum of $h_{k+1}(\cdot)$. However, condition (2.4a) (written with $\ell = \ell+1$) is not automatic in our setting, and the model may need to be properly adjusted. Indeed, even though

$$\begin{gathered}\psi_\ell(y) \le h_k(y),\\ \text{and } c^+(x^k) + \langle \hat{g}^i, y - x^k\rangle - \hat{\varepsilon}_i^k \le h_k(y),\ i \in \mathcal{B}_\ell^{agg},\end{gathered}$$

the same inequalities may not hold with h_k replaced by h_{k+1}. Specifically, if $f(x^k) < f(x^{k+1})$, which is possible, then we have that $h_k(y) \ge h_{k+1}(y)$. Thus, the key relations (2.4a)-(2.4c) are not guaranteed and in general do not hold.

There are various ways to ensure (2.4a)-(2.4c) after a serious step is taken. In fact, as discussed in [6], any approximation satisfying (2.4a) is acceptable, even a "bad" one, because the future null steps satisfying (2.4b) and (2.4c) would eventually build up a "good" approximation (of course, starting with a bad approximation is computationally inefficient). We next present one approach to ensure that all bundle elements correspond to appropriate approximate subgradients of the new function h_{k+1} at x^{k+1}, so that both convergence and computational efficiency are guaranteed. For oracle bundle elements, we only need to center (separate) linearization errors of f and c at the new point x^{k+1}. For the aggregate bundle elements, some special care is needed.

LEMMA 3.2. *Let ψ_ℓ be defined by* (3.4)*, using* (3.2) *and* (3.3)*. Suppose that the associated y^ℓ is declared a serious step, i.e., $x^{k+1} = y^\ell$. Then the following holds:*

(*i*) *For each $i \in \mathcal{B}_\ell^{oracle}$, the linearization errors*

$$\begin{aligned} e_{f_i}^{k+1} &= e_{f_i}^{k} + f(x^{k+1}) - f(x^k) + \langle g_f^i, x^k - x^{k+1}\rangle \\ e_{c_i}^{k+1} &= e_{c_i}^{k} + c(x^{k+1}) - c(x^k) + \langle g_c^i, x^k - x^{k+1}\rangle \end{aligned} \tag{3.5}$$

satisfy (3.1) *written with $k = k+1$. As a result, $g^i_{h_{k+1}} \in \partial_{e_i^{k+1}} h_{k+1}(x^{k+1})$, where $e_i^{k+1} \geq 0$ and $g^i_{h_{k+1}}$ are defined in* (3.3)*, written with k replaced by $k+1$.*

(*ii*) *For each $i \in \mathcal{B}_\ell^{agg}$, define*

$$\hat\varepsilon_i^{k+1} := \hat\varepsilon_i^{k} + c^+(x^{k+1}) - c^+(x^k) + \big(f(x^{k+1}) - f(x^k)\big)^+ + \langle \hat g^i, x^k - x^{k+1}\rangle. \tag{3.6}$$

Then $\hat\varepsilon_i^{k+1} \geq 0$ and $\hat g^i \in \partial_{\hat\varepsilon_i^{k+1}} h_{k+1}(x^{k+1})$.

Proof. Let $i \in \mathcal{B}_\ell^{oracle}$. Because $g_f^i \in \partial_{e_{f_i}^k} f(x^k)$, for all $y \in \mathbb{R}^n$ we have that

$$\begin{aligned} f(y) &\geq f(x^k) + \langle g_f^i, y - x^k\rangle - e_{f_i}^k \\ &= f(x^{k+1}) + \langle g_f^i, y - x^{k+1}\rangle \\ &\quad - \left(e_{f_i}^k + f(x^{k+1}) - f(x^k) + \langle g_f^i, x^k - x^{k+1}\rangle\right). \end{aligned}$$

Hence, $g_f^i \in \partial_{e_{f_i}^{k+1}} f(x^{k+1})$. By the same argument, $g_c^i \in \partial_{e_{c_i}^{k+1}} c(x^{k+1})$. Since f and c are convex, and $e_{f_i}^k$ and $e_{c_i}^k$ are nonnegative, (3.5) implies that $e_{f_i}^{k+1}, e_{c_i}^{k+1} \geq 0$. The remaining assertion of item (*i*) then follows by applying Lemma 3.1, where the quantities $(\ell, k, e_{f_i}^k, e_{c_i}^k)$ are replaced by $(\ell+1, k+1, e_{f_i}^{k+1}, e_{c_i}^{k+1})$, respectively.

Now, let $i \in \mathcal{B}_\ell^{agg}$. By (3.2), $\hat g^i \in \partial_{\hat\varepsilon_i^k} h_k(x^k)$. Hence, for all $y \in \mathbb{R}^n$, it holds that

$$h_k(y) \geq h_k(x^k) + \langle \hat g^i, y - x^k\rangle - \hat\varepsilon_i^k = c^+(x^k) + \langle \hat g^i, y - x^k\rangle - \hat\varepsilon_i^k. \tag{3.7}$$

In particular, for $y = x^{k+1}$, using the definitions of h_k and of $\hat\varepsilon_i^{k+1}$, (3.7) yields

$$\begin{array}{rcl} \max\{f(x^{k+1}) - f(x^k), c(x^{k+1})\} & \geq & c^+(x^k) + \langle \hat g^i, x^{k+1} - x^k\rangle - \hat\varepsilon_i^k \\ & = & -\hat\varepsilon_i^{k+1} + c^+(x^{k+1}) + \big(f(x^{k+1}) - f(x^k)\big)^+ . \end{array}$$

Thus, $\hat\varepsilon_i^{k+1} \geq c^+(x^{k+1}) + \big(f(x^{k+1}) - f(x^k)\big)^+ - \max\{f(x^{k+1}) - f(x^k), c(x^{k+1})\} \geq 0$. Now rewrite (3.7) as follows:

$$h_k(y) \quad \geq \quad c^+(x^{k+1}) + \langle \hat g^i, y - x^{k+1}\rangle - \left(\hat\varepsilon_i^k + c^+(x^{k+1}) - c^+(x^k) + \langle \hat g^i, x^k - x^{k+1}\rangle\right).$$

Using (3.6), we obtain that

$$h_k(y) \geq h_{k+1}(x^{k+1}) + \langle \hat{g}^i, y - x^{k+1}\rangle - \left(\hat{\varepsilon}_i^{k+1} - \big(f(x^{k+1}) - f(x^k)\big)^+\right). \tag{3.8}$$

The assertion of item *(ii)* would follow from (3.8), if we establish that

$$h_{k+1}(y) \geq h_k(y) - \big(f(x^{k+1}) - f(x^k)\big)^+ \text{ for all } y \in \mathbb{R}^n. \tag{3.9}$$

We proceed to prove (3.9).

If $f(x^{k+1}) \leq f(x^k)$, it easily follows that $h_{k+1}(y) \geq h_k(y)$ for all $y \in \mathbb{R}^n$. This obviously implies (3.9). Suppose now that $f(x^{k+1}) > f(x^k)$. For $y \in \mathbb{R}^n$ such that $f(y) - f(x^{k+1}) \geq c(y)$, we have that $h_{k+1}(y) - h_k(y) = -f(x^{k+1}) + f(x^k) = -\big(f(x^{k+1}) - f(x^k)\big)^+$, and so (3.9) holds. If $f(y) - f(x^{k+1}) < c(y)$ and $f(y) - f(x^k) \leq c(y)$, then $h_{k+1}(y) = h_k(y) = c(y)$, implying (3.9). Finally, if $f(y) - f(x^{k+1}) < c(y)$ and $f(y) - f(x^k) > c(y)$, we have that $h_{k+1}(y) > f(y) - f(x^{k+1}) = h_k(y) + f(x^k) - f(x^{k+1})$, which again gives (3.9).
The proof is complete. □

As a consequence of Lemma 3.2, regardless of whether the ℓ-th iteration produced a null step or a serious step, if

$$\mathcal{B}_{\ell+1}^{oracle} \subseteq \bigcup_{i \leq \ell} \Big\{ \big(f_i, c_i, e_{f_i}^{k+1}, e_{ci}^{k+1}, g_f^i, g_c^i\big) \Big\} \text{ and } \mathcal{B}_{\ell+1}^{agg} \subseteq \bigcup_{i \leq \ell} \Big\{ \big(\hat{\varepsilon}_i^{k+1}, \hat{g}^i\big) \Big\},$$

then the model

$$\psi_{\ell+1}(y) = c^+(x^k) + \max\Big\{ \max_{i \in \mathcal{B}_{\ell+1}^{oracle}} \{-e_i^k + \langle g_{h_k}^i, y - x^k\rangle\} \, , \max_{i \in \mathcal{B}_{\ell+1}^{agg}} \{-\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k\rangle\} \Big\},$$
$$k = k(\ell+1),$$

satisfies (2.4a) written with ℓ replaced by $\ell + 1$, with $h(\cdot) = h_k(\cdot)$, $k = k(\ell+1)$. Furthermore,
if $\big(\hat{\varepsilon}_\ell^k, \hat{g}^\ell\big) \subseteq \mathcal{B}_{\ell+1}^{agg}$, then $\psi_{\ell+1}$ satisfies (2.4b), and
if $\Big(f_\ell, c_\ell, e_{f_\ell}, e_{c\ell}, g_f^\ell, g_c^\ell\Big) \subseteq \mathcal{B}_{\ell+1}^{oracle}$, then $\psi_{\ell+1}$ satisfies (2.4c).

3.3. An Infeasible Constrained Proximal Bundle Method.

We are now in a position to give the algorithm in full details.

ALGORITHM 3.1 (Infeasible Constrained Proximal Bundle Method (ICPBM)).

Step 0. `Initialization.`
Choose parameters $m \in (0,1)$, $tol \geq 0$, and an integer $|\mathcal{B}|_{max} \geq 2$.
Choose $x^0 \in \mathbb{R}^n$. Set $y^0 := x^0$, and compute (f_0, c_0, g_f^0, g_c^0). Set $k = 0$, $\ell = 1$, $e_{f_0} := 0$, $e_{c0} := 0$ and define the starting bundles $\mathcal{B}_1^{oracle} := \{(e_{f_0}^0, e_{c0}^0, f_0, c_0, g_f^0, g_c^0)\}$ and $\mathcal{B}_1^{agg} := \emptyset$.

Step 1. `Quadratic Programming Subproblem.`
Choose $\mu_\ell > 0$ and compute y^ℓ as the solution to

$$\min_{y \in \mathbb{R}^n} \psi_\ell(y) + \frac{1}{2}\mu_\ell \|y - x^k\|^2 \,, \tag{3.10}$$

```
          where $\psi_\ell$ is defined by (3.4) and (3.3). Compute
              $\hat{g}^\ell = \mu_\ell(x^k - y^\ell), \quad \hat{\varepsilon}_\ell^k = c^+(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell}\|\hat{g}^\ell\|^2, \quad \delta_\ell = \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2.$
          Compute $(f_\ell, c_\ell, g_f^\ell, g_c^\ell)$ and $(e_{f_\ell}^k, e_{c\ell}^k)$, using (3.1) written with $i = \ell$.
Step 2.   Stopping test.
          If $\delta_\ell \le tol$, stop.
Step 3.   Descent test.   Compute $h_\ell := h_k(y^\ell) = \max\{f_\ell - f(x^k), c_\ell\}$.
          If $h_\ell \le c^+(x^k) - m\delta_\ell$, then declare a serious step. Otherwise, declare a null step.
Step 4.   Bundle Management.
          Set $\mathcal{B}_{\ell+1}^{oracle} := \mathcal{B}_\ell^{oracle}$ and $\mathcal{B}_{\ell+1}^{agg} := \mathcal{B}_\ell^{agg}$.
          If the bundle has reached the maximum bundle size, i.e.,
          if $|\mathcal{B}_{\ell+1}^{oracle} \cup \mathcal{B}_{\ell+1}^{agg}| = |\mathcal{B}|_{max}$, then:
          Delete at least two elements from $\mathcal{B}_{\ell+1}^{oracle} \cup \mathcal{B}_{\ell+1}^{agg}$.
          Insert the aggregate couple $(\hat{\varepsilon}_\ell^k, \hat{g}^\ell)$ in $\mathcal{B}_{\ell+1}^{agg}$.
          Append $(e_{f_\ell}^k, e_{c\ell}^k, f_\ell, c_\ell, g_f^\ell, g_c^\ell)$ to $\mathcal{B}_{\ell+1}^{oracle}$.
Step 5.   Model adjustment (serious step).
          If $y^\ell$ is a serious step, then:
          Define the next stability center: $\big(x^{k+1}, f(x^{k+1}), c(x^{k+1})\big) := \big(y^\ell, f_\ell, c_\ell\big)$.
          Update the linearization errors for $i \in \mathcal{B}_{\ell+1}^{oracle}$ using (3.5) in Lemma 3.2.
          Update the aggregate errors for $i \in \mathcal{B}_{\ell+1}^{agg}$ using (3.6) in Lemma 3.2.
          Set $k = k + 1$.
  Loop.   Set $\ell = \ell + 1$ and go to Step 1.
```

Some remarks are in order. Recalling the definition of $h_k(\cdot)$, we conclude that if the descent test is satisfied and a serious step is declared, then it must hold that

$$f(x^{k+1}) - f(x^k) \le c^+(x^k) - m\delta_\ell\,, \tag{3.11}$$

and

$$c(x^{k+1}) \le c^+(x^k) - m\delta_\ell\,. \tag{3.12}$$

In particular, if x^k is infeasible, then $f(x^{k+1}) > f(x^k)$ is possible (since $c^+(x^k) > 0$). Therefore, the method is not monotone with respect to f when outside of the feasible region. However, outside of the feasible region it is monotone with respect to c, because $c(x^{k+1}) < c^+(x^k) = c(x^k)$ for x^k infeasible. This seems intuitively reasonable: while it is natural to accept the increase in the objective function value in order to decrease infeasibility, it is not so clear why one would want to decrease the objective function at the expense of moving away from the feasible region. The situation reverses when x^k is feasible. In that case, $c^+(x^k) = 0$, so that $f(x^{k+1}) < f(x^k)$. But although (3.12) implies that x^{k+1} is feasible too, it is possible that $c(x^{k+1}) > c(x^k)$ (except when $c(x^k)$ is exactly zero). This also appears completely reasonable: while preserving feasibility, we allow c to increase (so that the boundary of the feasible set can be approached), at the same time obtaining a decrease in the objective function.

In Algorithm 3.1, we do not specify any rule for choosing the proximal parameter μ_ℓ. Conditions that μ_ℓ should satisfy for convergence are very mild, and they are stated in the convergence results of Section 4. That said, a sound strategy for choosing this parameter is important for computational efficiency. Actually, this is yet another advantage of having our development follow closely the well-established and well-tested unconstrained bundle methods: we can use the update rules for the former, which are already known to perform well in practice, e.g., [21, 25, 34].

Subproblem (3.10) is handled by solving its equivalent quadratic programming formulation

$$c^+(x^k) + \left\{ \begin{array}{cl} \min\limits_{(y,t)\in\mathbb{R}^{n+1}} & t + \frac{1}{2}\mu_\ell\|y - x^k\|^2 \\ \text{s.t.} & -e_i^k + \langle g_{h_k}^i, y - x^k\rangle \le t\,,\ i \in \mathcal{B}_\ell^{oracle}\,, \\ & -\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k\rangle \le t\,,\ i \in \mathcal{B}_\ell^{agg}\,, \end{array} \right. \tag{3.13}$$

or the dual of this problem. The dual of (3.13) can be written as a quadratic programming problem on the unit simplex, for which specialized highly effective methods are available, e.g., [18, 9]. The number of variables in the latter is precisely the number of elements in the bundle, which shows the importance of Step 4 of Algorithm 3.1.

The following well-known characterization of the solution of (3.10) follows from [2, Lemma 9.8]. Those relations have already been discussed in Section 2, but here we state them in the notation of Algorithm 3.1.

LEMMA 3.3. *In the setting of Algorithm 3.1, it holds that*

(i) $y^\ell = x^k - \frac{1}{\mu_\ell}\hat{g}^\ell$*, where* $\hat{g}^\ell \in \partial\psi_\ell(y^\ell)$*.*

(ii) $\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k)$*, where* $\hat{\varepsilon}_\ell^k \ge 0$*.*

In particular, it follows that $\delta_\ell \ge 0$ in Algorithm 3.1. Moreover, if $\delta_\ell = 0$ for some k, then $\hat{\varepsilon}_\ell^k = 0$ and $\hat{g}^\ell = 0$. Hence, $0 \in \partial h_k(x^k)$ and x^k is a solution to (1.1), by Theorem 2.1.

4. Convergence results.

We assume from now on that the stopping tolerance *tol* is set to zero, $\delta_\ell > 0$ for all ℓ, and thus Algorithm 3.1 does not terminate and generates an infinite sequence of iterates. As usual in the convergence analysis of bundle methods, we consider the following two possible cases: the number of serious steps is either infinite or finite (in the second case, the last generated serious step is followed by an infinite number of null steps).

In the sequel, D denotes the feasible set of (1.1), i.e.,

$$D := \{x \in \mathbb{R}^n \mid c(x) \le 0\}.$$

Given an index k of a serious step, let $\ell(k)$ be the index of y^ℓ which produced this serious step, i.e., $y^{\ell(k)} = x^k$. Finally, the set $\mathcal{L}_s := \{\ell \mid y^\ell \text{ is a serious step}\}$ collects the indices of serious steps in the sequence $\{y^\ell\}$.

PROPOSITION 4.1. *For any serious iteration index* $k_0 \ge 0$*, it holds that*

$$x^k \in \{x \in \mathbb{R}^n \mid c(x) \le c^+(x^{k_0})\} \quad \text{for all } k \ge k_0\,.$$

In particular, if $x^{k_1} \in D$ *for some* $k_1 \ge 0$*, then* $x^k \in D$ *for all* $k \ge k_1$*.*

Proof. Fix an arbitrary $k_0 \ge 0$. If k_0 is the last serious step (i.e., all the subsequent steps are declared null), then the first assertion is trivial.

Suppose now that there exists the (k_0+1)-st serious step. If $x^{k_0} \notin D$, then (3.12) implies that $c(x^{k_0+1}) < c^+(x^{k_0}) = c(x^{k_0})$. If further $x^k \notin D$ for all $k \ge k_0$, then repeating the above argument we conclude that the sequence $\{c(x^k)\}$ is nonincreasing. In particular, $c(x^k) \le c(x^{k_0}) = c^+(x^{k_0})$ for all $k \ge k_0$.

Suppose now that $x^{k_1} \in D$ for some k_1. If k_1 is the last serious step, the second assertion is trivial. If there exists the (k_1+1)-st serious step, then (3.12) implies that

$c(x^{k_1+1}) \leq -m\delta_{\ell(k_1+1)} < 0$. Using (3.12) recursively, we conclude that $c(x^k) < 0 = c^+(x^{k_1})$ for all $k > k_1$, i.e., $x^k \in D$. Thus the second assertion holds.

Noting that $c(x^k) < c(x^{k_0}) = c^+(x^{k_0})$ for $k_0 \leq k \leq k_1$, and $c(x^k) < 0 \leq c^+(x^{k_0})$ for $k > k_1$, concludes the proof. □

PROPOSITION 4.2. *Let f be bounded below on D, and suppose that Algorithm 3.1 generates an infinite number of serious steps. Then $\{\hat{\varepsilon}_\ell^k\}_{\ell \in \mathcal{L}_s} \to 0$. Furthermore,*

(i) If

$$\sum_{\ell \in \mathcal{L}_s} \frac{1}{\mu_\ell} = +\infty\,, \tag{4.1}$$

then zero is an accumulation point of the sequence $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s}$.

(ii) If for some $\bar{\mu} > 0$ it holds that

$$\mu_\ell \leq \bar{\mu},\ \ell \in \mathcal{L}_s\,, \tag{4.2}$$

then $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s} \to 0$.

Proof. We first show that

$$\sum_{\ell \in \mathcal{L}_s} \delta_\ell < +\infty\,. \tag{4.3}$$

By Proposition 4.1, either $x^k \notin D$ for all k, or there exists some index k_1 such that $x^k \in D$ for all $k \geq k_1$. We examine the two possibilities separately.
In the first case, (3.12) gives that

$$m\delta_{\ell(k+1)} \leq c(x^k) - c(x^{k+1}), \quad k \geq 0\,. \tag{4.4}$$

Thus, the sequence $\{c(x^k)\}$ is decreasing, and since $x^k \notin D$ for all k, this sequence is bounded below (by zero). It follows that it converges to some $\bar{c} \geq 0$ and, furthermore, that $c(x^k) \geq \bar{c}$ for all k. Therefore, summing up the relation (4.4) over all $\ell \in \mathcal{L}_s$, we obtain that

$$\sum_{\ell \in \mathcal{L}_s} \delta_\ell \leq \frac{c(x^0) - \bar{c}}{m}.$$

Consider now the second case, i.e., $x^k \in D$ for $k \geq k_1$ (and let k_1 be the first index such that $x^{k_1} \in D$). Then (3.11) yields

$$m\delta_{\ell(k+1)} \leq f(x^k) - f(x^{k+1})\,, \quad k \geq k_1\,. \tag{4.5}$$

Hence, the sequence $\{f(x^k)\}_{k \geq k_1}$ is decreasing and bounded below by $\bar{f} = \inf\{f(x) \mid x \in D\}$. Recall that for $k < k_1$, $x^k \notin D$, and thus (4.4) holds. Summing up the relations in (4.4) and (4.5), we obtain that

$$\sum_{\ell \in \mathcal{L}_s} \delta_\ell = \sum_{\ell \in \mathcal{L}_s,\, \ell < \ell(k_1)} \delta_\ell + \sum_{\ell \in \mathcal{L}_s,\, \ell \geq \ell(k_1)} \delta_\ell \leq \frac{1}{m}(c(x^0) - c(x^{k_1-1}) + f(x^{k_1}) - \bar{f})\,.$$

This completes the proof of (4.3).

By the definition of δ_ℓ in Algorithm 3.1, for all ℓ it holds that

$$\frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2 \leq \delta_\ell \quad \text{and} \quad \hat{\varepsilon}_\ell^k \leq \delta_\ell\,. \tag{4.6}$$

By (4.3), $\{\delta_\ell\}_{\ell\in\mathcal{L}_s} \to 0$. It immediately follows that $\{\hat{\varepsilon}_\ell^k\}_{\ell\in\mathcal{L}_s} \to 0$. If (4.2) holds, clearly also $\{\hat{g}^\ell\}_{\ell\in\mathcal{L}_s} \to 0$.

To prove item (i), suppose that the sequence $\{\|\hat{g}^\ell\|\}_{\ell\in\mathcal{L}_s}$ is bounded away from zero. Then, by (4.6) and (4.1), we obtain that $\sum_{\ell\in\mathcal{L}_s}\delta_\ell = +\infty$, in contradiction with (4.3). □

We next exhibit the conditions under which the serious iterates are bounded.

PROPOSITION 4.3. *Suppose that problem* (1.1) *has a solution* $\bar{x}$ *and that Algorithm 3.1 generates an infinite sequence of serious steps. Then the sequence* $\{x^k\}$ *is bounded if any of the following conditions is satisfied:*

(i) *the feasible set* D *is bounded,*

or

(ii) *there exists some iteration index* k_1 *such that* $f(\bar{x}) \le f(x^k)+c^+(x^k)$ *for all* $k \ge k_1$ *(in particular, this is true if* $x^{k_1} \in D$ *for some* k_1*),and* $\mu_\ell \ge \hat{\mu}$, $\ell \in \mathcal{L}_s$, *for some* $\hat{\mu} > 0$.

Proof. Since D is a level set of c, if (i) holds then the convexity of c implies that all the level sets of c are bounded. Boundedness of $\{x^k\}$ now follows from the first assertion of Proposition 4.1.

Suppose now that (ii) holds. (Observe that if $x^{k_1} \in D$ for some k_1, then $x^k \in D$ for all $k \ge k_1$, by Proposition 4.1. In that case, $f(\bar{x}) \le f(x^k) = f(x^k) + c^+(x^k)$ holds automatically). For $\ell = \ell(k+1)$, we have that

$$\begin{aligned}
\|x^{k+1}-\bar{x}\|^2 &= \|x^k-\bar{x}\|^2 - \frac{2}{\mu_\ell}\langle \hat{g}^\ell, x^k-\bar{x}\rangle + \frac{1}{\mu_\ell^2}\|\hat{g}^\ell\|^2 \\
&\le \|x^k-\bar{x}\|^2 + \frac{2}{\mu_\ell}(h_k(\bar{x}) - h_k(x^k) + \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2) \\
&= \|x^k-\bar{x}\|^2 + \frac{2}{\mu_\ell}(h_k(\bar{x}) - h_k(x^k) + \delta_\ell)\,,
\end{aligned} \tag{4.7}$$

where we have used that $x^{k+1}-x^k = y^\ell - x^k = \hat{g}^\ell/\mu_\ell$ and $\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k)$ (see Lemma 3.3), and the definition of δ_ℓ in Algorithm 3.1.

Observe further that

$$h_k(\bar{x}) - h_k(x^k) = \max\bigl\{f(\bar{x}) - f(x^k), c(\bar{x})\bigr\} - c^+(x^k).$$

The quantity above is clearly nonpositive if $f(\bar{x})-f(x^k)-c^+(x^k) \le 0$. This inequality is ensured by the second condition in (ii), for all $k \ge k_1$. In that case, (4.7) (using also that $\mu_\ell \ge \hat{\mu} > 0$) yields

$$\|x^{k+1}-\bar{x}\|^2 \le \|x^k-\bar{x}\|^2 + \frac{2}{\hat{\mu}}\delta_\ell\,, \quad k \ge k_1\,, \ell = \ell(k+1) \in \mathcal{L}_s\,. \tag{4.8}$$

By (4.3) and [33, Lemma 2, p.44], we conclude that the sequence $\{\|x^{k+1}-\bar{x}\|\}$ converges. Hence, the sequence $\{x^k\}$ is bounded. □

The assumption that the feasible set of (1.1) is bounded was also imposed in [28, 19, 22, 24, 7]. According to Proposition 4.3, we do not need this assumption if the iterates enter the feasible region. Methods in [16] are all feasible, except for the “phase I-phase II” modifications briefly sketched in [16, Ch. 5.7]. The main convergence result therein is [16, Thm. 5.7.4], which does not assume boundedness of the feasible set, but also does not establish the existence of accumulation points for infeasible sequences of serious steps. Rather, the analysis concerns properties of accumulation points, without claiming their existence.

We next present the final convergence result for the case of the infinite number of serious steps.

THEOREM 4.4. *Assume that* (1.1) *satisfies the Slater constraint qualification, and that its solution set is nonempty. Suppose that Algorithm 3.1 generates an infinite sequence of serious steps, which is bounded (this holds, for example, under any of the two assumptions of Proposition 4.3).*

If condition (4.1) *holds, then the sequence* $\{x^k\}$ *has an accumulation point which is a solution to* (1.1).

If condition (4.2) *holds, then all the accumulation points of* $\{x^k\}$ *are solutions to* (1.1).

If either (4.1) *or* (4.2) *holds, then in the setting of Proposition 4.3*(*ii*)*, the whole sequence* $\{x^k\}$ *converges to a solution to* (1.1).

Proof. Fix an arbitrary $y \in \mathbb{R}^n$. By Lemma 3.3, for any serious step index k, it holds that

$$h_{x^k}(y) \geq c^+(x^k) + \langle \hat{g}^\ell, y - x^k \rangle - \hat{\varepsilon}_\ell^k\,, \quad \ell = \ell(k) \in \mathcal{L}_s. \tag{4.9}$$

If (4.1) holds, there exists a subsequence of $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s}$ converging to zero (by Proposition 4.2). Also, $\{\hat{\varepsilon}_\ell^k\}_{\ell \in \mathcal{L}_s} \to 0$. Since $\{x^k\}$ is bounded, taking a further subsequence (if necessary), and passing onto the limit in (4.9), we obtain that

$$h_{\bar{x}}(y) \geq c^+(\bar{x}) + \langle 0, y - \bar{x} \rangle - 0 = c^+(\bar{x}) = h_{\bar{x}}(\bar{x})\,,$$

where $\bar{x}$ is an accumulation point of $\{x^k\}$. Since $y \in \mathbb{R}^n$ is arbitrary, the above means that

$$\min\{h_{\bar{x}}(y) \mid y \in \mathbb{R}^n\} = h_{\bar{x}}(\bar{x}) = c^+(\bar{x})\,. \tag{4.10}$$

According to Theorem 2.1, it remains to prove that

$$h_{\bar{x}}(\bar{x}) = c^+(\bar{x}) = 0\,.$$

If $c^+(\bar{x}) > 0$, by continuity it holds that $c^+(y) = c(y) > f(y) - f(\bar{x})$ for all y in some neighborhood of $\bar{x}$. Hence, in such a neighborhood $h_{\bar{x}}(y) = c^+(y)$. It follows from (4.10) that $c^+(\cdot)$ has a local minimum at $\bar{x}$, with $c^+(\bar{x}) > 0$. Since $c^+(\cdot)$ is convex, this minimum must be also global, which contradicts the fact that $D = \{x \in \mathbb{R}^n \mid c^+(x) = 0\} \neq \emptyset$.

If (4.2) holds, then $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s} \to 0$, and we can repeat the above argument by passing onto the limit along any convergent subsequence of $\{x^k\}$.

Finally, in the setting of Proposition 4.3(*ii*), we can choose $\bar{x}$ in (4.8) as an accumulation point of $\{x^k\}$ which is a solution to (1.1). Then $\{\|x^k - \bar{x}\|\}$ converges. Since it has a subsequence converging to zero, it must be the case that $\{x^k\} \to \bar{x}$. □

We conclude by considering the case when the number of serious steps is finite, i.e., there exists $\ell ast = \max\{\ell \mid \ell \in \mathcal{L}_s\}$. We denote the corresponding last serious iteration by $k_{\ell ast}$. Then the function $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$ is fixed from that point on, and the algorithm generates null steps only. The fact that $x^{k_{\ell ast}}$ is a solution to (1.1) can be proved similarly to standard results on bundle methods, e.g., [6]. Note however that, unlike [6], we do not assume that the proximal parameter is fixed after the last serious step.

THEOREM 4.5. *Assume that* (1.1) *satisfies the Slater constraint qualification. Suppose that Algorithm 3.1 makes a finite number of serious steps. If $\bar{\mu} \geq \mu_{\ell+1} \geq \mu_\ell$ for all $\ell \geq \ell ast$, then $x^{k_{\ell ast}}$ is a solution to* (1.1).

Proof. In the sequel, we consider $\ell \geq \ell ast$ and denote $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$. Observe first that for any $y \in \mathbb{R}^n$,

$$\begin{aligned} l_{k,\ell}(y) &= h(x^{k_{\ell ast}}) - \hat{\varepsilon}_\ell^k + \langle \hat{g}^\ell, y - x^{k_{\ell ast}} \rangle \\ &= \psi_\ell(y^\ell) + \mu_\ell \langle x^{k_{\ell ast}} - y^\ell, y - y^\ell \rangle. \end{aligned}$$

In particular, $l_{k,\ell}(y^\ell) = \psi_\ell(y^\ell)$, and further,

$$l_{k,\ell}(y) + \frac{1}{2}\mu_\ell \|y - x^{k_{\ell ast}}\|^2 = \psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 + \frac{1}{2}\mu_\ell \|y - y^\ell\|^2, \quad y \in \mathbb{R}^n. \tag{4.11}$$

We have that

$$\begin{aligned} h(x^{k_{\ell ast}}) &\geq \psi_{\ell+1}(x^{k_{\ell ast}}) \\ &\geq \psi_{\ell+1}(y^{\ell+1}) + \frac{1}{2}\mu_{\ell+1} \|y^{\ell+1} - x^{k_{\ell ast}}\|^2 \\ &\geq l_{k,\ell}(y^{\ell+1}) + \frac{1}{2}\mu_\ell \|y^{\ell+1} - x^{k_{\ell ast}}\|^2 \\ &= \psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 + \frac{1}{2}\mu_\ell \|y^{\ell+1} - y^\ell\|^2, \end{aligned} \tag{4.12}$$

where the first inequality is by (2.4a), the second inequality is by the definition of $y^{\ell+1}$, the third is by (2.4b) and $\mu_{\ell+1} \geq \mu_\ell$, and the equality is by (4.11).

It follows from (4.12) that the sequence $\{\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2\}$ is nondecreasing and bounded above. Hence, it converges. Fixing $y = x^{k_{\ell ast}}$ in (4.11), we have that

$$h(x^{k_{\ell ast}}) \geq l_{k,\ell}(x^{k_{\ell ast}}) = \left(\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 \right) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2,$$

where the inequality is by (2.4a), (2.4b). Since $\{\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2\}$ converges, it follows that $\{y^\ell\}$ is bounded. Also, since $\{\mu_\ell\}$ is bounded below by $\mu_{\ell ast} > 0$, (4.12) implies that

$$\{y^{\ell+1} - y^\ell\} \to 0, \ \ell \to \infty. \tag{4.13}$$

Since $\{y^\ell\}$ is bounded, the convex function h can be considered to be Lipschitz-continuous (on the bounded set of interest), and we further have that $\{\hat{g}^\ell\}$ is bounded on that set. Hence,

$$\begin{aligned} L\|y^{\ell+1} - y^\ell\| &\geq h(y^{\ell+1}) - h(y^\ell) \\ &\geq \psi_{\ell+1}(y^{\ell+1}) - h(y^\ell) \\ &\geq \langle \hat{g}^\ell, y^{\ell+1} - y^\ell \rangle, \end{aligned}$$

where the second inequality is by (2.4a) and the third is by (2.4c). Thus (4.13) implies that

$$\{h(y^\ell) - \psi_{\ell+1}(y^{\ell+1})\} \to 0, \ \ell \to \infty. \tag{4.14}$$

Let $\bar{y}$ be any accumulation point of $\{y^\ell\}$, i.e., $\{y^{\ell_i}\} \to \bar{y}$ as $i \to \infty$. Note that by (4.13), we have that $\{y^{\ell_i - 1}\} \to \bar{y}$. Then (4.14) and the continuity of h imply that

$$\{\psi_{\ell_i}(y^{\ell_i})\} \to h(\bar{y}),\ i \to \infty. \tag{4.15}$$

Moreover, for any $y \in \mathbb{R}^n$, we have that

$$h(y) \geq \psi_\ell(y) \geq \psi_\ell(y^\ell) + \langle \hat{g}^\ell, y - y^\ell \rangle = \psi_\ell(y^\ell) + \mu_\ell \langle x^{k_{\ell ast}} - y^\ell, y - y^\ell \rangle,$$

where the first inequality is by (2.4a) and the other relations are by Lemma 3.3. Passing onto the limit along the specified subsequence as $i \to \infty$, and using (4.15), we conclude that

$$h(y) \geq h(\bar{y}) + \tilde{\mu} \langle x^{k_{\ell ast}} - \bar{y}, y - \bar{y} \rangle, \quad y \in \mathbb{R}^n,$$

where $\tilde{\mu}$ is the limit of the (nondecreasing and bounded above) sequence $\{\mu_\ell\}$. It follows that

$$\tilde{\mu}(x^{k_{\ell ast}} - \bar{y}) \in \partial h(\bar{y}), \tag{4.16}$$

or equivalently,

$$\bar{y} \text{ is the solution to } \min_{y \in \mathbb{R}^n} h(y) + \frac{1}{2}\tilde{\mu}\|y - x^{k_{\ell ast}}\|^2.$$

In particular, the latter means that

$$h(\bar{y}) + \frac{1}{2}\tilde{\mu}\|\bar{y} - x^{k_{\ell ast}}\|^2 \leq h(x^{k_{\ell ast}}). \tag{4.17}$$

On the other hand, since the descent test never holds for $\ell \geq \ell ast$, we obtain that

$$\begin{array}{rcl} h(y^\ell) - h(x^{k_{\ell ast}}) & > & -m\delta_\ell \\ & = & -m\left(h(x^{k_{\ell ast}}) - \psi_\ell(y^\ell) - \frac{1}{2}\mu_\ell\|y^\ell - x^{k_{\ell ast}}\|^2\right) \\ & \geq & -m\left(h(x^{k_{\ell ast}}) - \psi_\ell(y^\ell)\right). \end{array}$$

Passing onto the limit along the specified subsequence as $i \to \infty$, and using (4.15), we obtain that

$$0 \geq (1 - m)\left(h(x^{k_{\ell ast}}) - h(\bar{y})\right).$$

As $m \in (0,1)$, we have that $h(x^{k_{\ell ast}}) \leq h(\bar{y})$. But then (4.17) implies that $\bar{y} = x^{k_{\ell ast}}$. Recalling (4.16), we have that $0 \in \partial h(x^{k_{\ell ast}})$, where $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$. By Theorem 2.1, $x^{k_{\ell ast}}$ is a solution to (1.1). □

5. Preliminary computational experience. For our numerical assessment, we use the following set of academic problems:

- `CHAIN`, a problem that minimizes a linear function over a piecewise quadratic constraint set. The physical interpretation of this problem is to find the equilibrium of a bidimensional chain formed by 20 links. The chain has end points fixed at the coordinates $(0,0)$ and $(1,0)$, the length of each link should be less than 0.10; see [24, p. 146]. To choose the starting point, we consider two chains lying on the horizontal axis with endpoints as above, but different lenghts:

Feasible: 20 identical links, each one of length 0.10. (5.1a)

Infeasible: 20 identical links, each one of length 0.12. (5.1b)

– `MAXQUAD`, the piecewise quadratic objective function is taken from [23, p. 151], and the constraint is given by $c(x) = \max\left\{\max_{i=1,\dots,10}|x_i| \leq 0.05, \sum_{i=1}^{10} x_i \leq 0.05\right\}$.

– `LOCAT`, a minimax location problem of dimension 4 with the objective function given by the maximum of weighted normed functions, and a piecewise quadratic constraint, [3].

– `MINSUM`, a minsum location problem of dimension 6, with the objective function given by a weighted sum of norms, and a linear constraint, [3].

– `ROSEN`, the Rosen-Susuki problem from [14, p. 66]. It has dimension 4, solution $\bar{x} = (0, 1, 2, -1)$, $f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$, and

$$c(x) = \max\left\{\begin{array}{l} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \\ x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \\ x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \end{array}\right\}.$$

– `HILBERT`, a Hilbert-like feasibility problem of dimension n. For $f(x) \equiv 0$, the constraint is given by $c(x) = \max_{i \leq n}\left\{\max_{k \leq n} |\sum_{=1}^{n} \frac{x_j - 1}{i+k+j-2}|\right\}$, so $\bar{x} = (1, \dots, 1)$ is the solution.

Table 5.1 below shows some additional relevant data for the problems, including the dimensionality, optimal value and starting points. For each of the problems we used two starting points: feasible and infeasible. The exception is `HILBERT`, which is a feasibility problem, and so only an infeasible starting point is of interest.

Name	n	$f(\bar{x})$	Feasible x^0	Infeasible x^0
`CHAIN`	38	-9.103962328	see (5.1a)	see (5.1b)
`MAXQUAD`	10	-0.368166	$x_i = 0$	$x_i = 10$
`LOCAT`	4	23.88676767	$(15, 22, 26, 11)$	$x_i = 10$
`MINSUM`	6	68.82956	$(0, 0, 0, 0, 3, 0)$	$x_i = 10$
`ROSEN`	4	-44	$x_i = 0$	$(-1, 2, -3, -4)$
`HILBERT`	50	0	–	$x_i = 10$

TABLE 5.1
Some problem data

Since all these problems have known optimal values, the exact improvement function $h_{\bar{x}}$ is available. For comparison purposes, we first solve the unconstrained problem of minimizing $h_{\bar{x}}$ using N1CV2, the proximal bundle method for unconstrained problems, described in [25] (with QP subproblems solved by the method described in [18]) and available upon request at
`www-rocq.inria.fr/estime/modulopt/optimization-routines/n1cv2.html`.
These runs can be thought of as an ideal situation, in which the constrained optimization problem (1.1) is replaced by a single (equivalent) unconstrained problem. The obtained results can therefore be used as a benchmark for ICPBM, whose implementation was built on top of N1CV2, and in particular, employs the same warm start-ups and update rules for all parameters, including the crucial update of μ_ℓ.

All runs were performed on a Pentium II 400MHz with 128Mb RAM. The size of the bundle was limited to 100 elements. Optimality is declared when

$$\hat{\varepsilon}_\ell^k \leq 10^{-4} \quad \text{and} \quad \|\hat{g}^\ell\|^2 \leq 10^{-8}.$$

We note that the above split stopping criterion is generally preferable to the one based on δ_ℓ, because the split criterion does not depend on μ_ℓ.

Our numerical results are reported in Table 5.2 below. For each run, and for both algorithms, we give the total number of iterations (i.e., calls to the oracle) and the final accuracy with respect to the (known) optimal value of the problem (i.e., the number of exact digits in the final objective function value). In all cases, the final value of the constraint obtained with ICPBM was less than 10^{-4}.

	CHAIN		MAXQUAD		LOCAT		MINSUM		ROSEN		HILBERT	
	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc
N1CV2 x^0 feas.	112	4	65	4	22	6	57	6	20	6	-	-
N1CV2 x^0 infeas.	167	4	97	5	33	6	63	6	22	6	31	7
ICPBM x^0 feas.	210	5	160	6	19	6	86	5	38	5	-	-
ICPBM x^0 infeas.	361	4	241	5	30	6	115	5	37	5	29	5

TABLE 5.2
Summary of results

In our opinion, Table 5.2 shows a reasonable performance of ICPBM. In all the cases, the method succeeds in obtaining a reasonably high accuracy, at the price of less than three times the number of oracle calls required by N1CV2 to solve the "ideal" unconstrained problem of minimizing $h_{\bar{x}}$. Furthermore, the results for HILBERT (whose objective function is constant) confirm that the two codes are about as efficient when solving a problem of the same complexity (in that case, in some sense unconstrained). With respect to the influence of starting points, ICPBM's behaviour does not seem much affected by the choice of an infeasible x^0. The slowest convergence is observed in CHAIN and MAXQUAD. We conjecture that some nasty bound interferes in these problems (a similar behaviour is observed for N1CV2).

In our opinion, comparing our numerical results with those obtained by other authors is problematic, even if some of the test problems are seemingly the same. First of all, when discussing numerical results in NSO, an important issue arises which is sometimes referred to as the "curse of nondifferentiability". Namely, because of discontinuity of the subdifferential, even the same code can produce very different output when running on different computational platforms, see [2, p. 102]. This phenomenon, together with the lack of a standard universally accepted NSO problems library, makes broad numerical comparisons difficult. Thus some caution should be exercised when making the conclusions. Nevertheless, the limited experience reported above suggests that the approach presented in this paper is computationally viable. But to be fair, we should mention that our results appear worse than those reported in [22] for some of the same problems. However, we note that even our results for minimizing the fixed "ideal" unconstrained function $h_{\bar{x}}$ by N1CV2 are worse than what is reported in [22]. For example, while in average the five algorithms in [22] solve MAXQUAD in 33/42 iterations (for feasible/infeasible starting point, respectively), N1CV2 needs 65/97 iterations to minimize $h_{\bar{x}}$. We do not have an explanation, as this could be caused by various implementational differences, all secondary to the ideas of the respective methods themselves: rules for choosing the proximal parameter, linesearch rules, bundle selection rules, oracle rules for choosing subgradients, treating linear constraints and bounds inside or outside of QP subproblems, etc.

In particular, one implementational difference, which can be significant, is that in [22] some linear constraints are inserted into the QP subproblems, while we do not make a distinction between linear and nonlinear constraints. In fact, our results that compare better to the results in [22] are precisely the two problems which do not have any linear constraints, i.e., `LOCAT` and `ROSEN` (19/30 and 38/37 iterations for ICPBM versus 12/15 and 22/30 iterations in average for the five algorithms in [22]). For this reason, we made a more thorough study of the performance of ICPBM on these two problems. Specifically, we analyze if the algorithm is following closely the (curved) boundary of the feasible set, a behaviour that is known to prevent fast convergence. Figures 5.1 and 5.2 below show, respectively for `LOCAT` and `ROSEN`, the (last iterates of the) constraint values generated by ICBPM, starting from feasible and infeasible points.

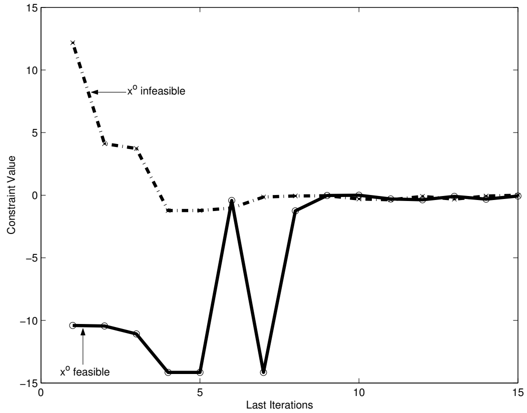

FIG. 5.1. *Constraint values for* `LOCAT`

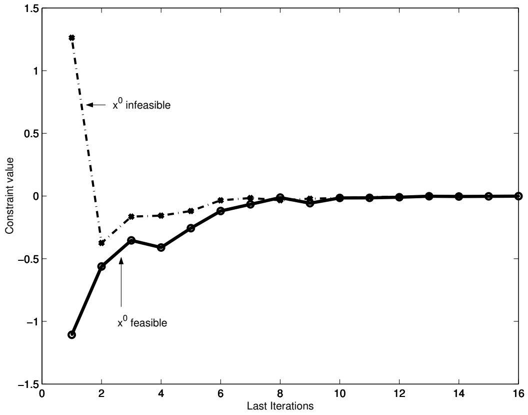

FIG. 5.2. *Constraint values for* `ROSEN`

In Table 5.2 we see that the faster convergence is achieved for `LOCAT`, which, as shown in Figure 5.1, is not generating iterates close to the boundary of the feasible set. By contrast, the phenomenon does appear in ROSEN: note that the scale in the vertical axis of Figure 5.1 is 10 times bigger than in Figure 5.2.

We next analyze the effect of constraint scaling, which is a general concern in constrained optimization. We ran ICPBM on 9 instances of the test problem LOCAT, with the constraints multiplied by a factor ranging in $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$. We kept the same stopping tolerances as for the unscaled problem and obtained, for all instances, between 6 and 9 digits of accuracy, with (unscaled) constraint values of the order of 10^{-3} or better.

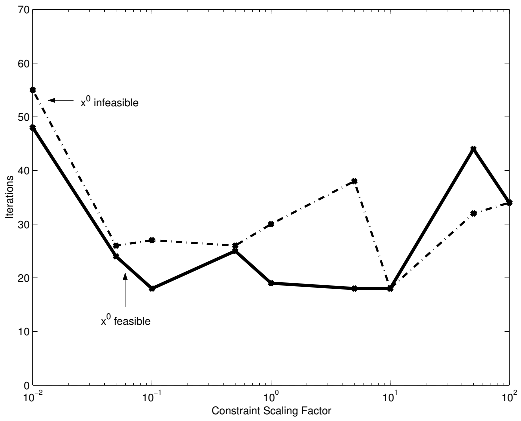

FIG. 5.3. *The effect of scaling*

Figure 5.3 shows, for both feasible and infeasible starting points, ICPBM's total number of iterations in relation to the scaling factor, displayed in the semilogarithmic scale. As expected, the number of iterations increases as the factor gets bigger. However, especially for the infeasible starting point, the overall behaviour of the algorithm is not dramatically changed. More precisely, the average of iterations for the 9 instances with feasible starting point is 28, about a 50% increase over the 19 iterations for the unscaled case in Table 5.2. By contrast, the infeasible starting point, that could be thought of as more difficult, gave an average number of iterations equal to 32, versus 30 iterations needed for the unscaled case in Table 5.2.

To conclude, we note that in [30, Sec. 5] a bundle algorithm for one-dimensional problems is presented, where an appropriate modification in the definition of linearization errors makes directions independent of constraint scaling. Those ideas are also valid in $\mathbb{R}^n$, and therefore could be incorporated in ICPBM, if scaling is of concern.

6. Concluding Remarks. We have presented a new idea for handling constraints in nonsmooth convex minimization. Among the features of this approach which can be useful, we mention the following:

- the method can start from infeasible points;
- the method does not use penalty functions, and thus does not require estimating a suitable value of penalty parameter;
- the method does not use complex filter technologies;
- the method is close in spirit and structure to standard unconstrained bundle methods, and thus can build on the available software and theory (e.g., aggregation and compression techniques);

– convergence is established under mild assumptions.

Our preliminary numerical results show the viability of the method, although implementational improvements are both possible and necessary.

An interesting subject of future research can be an extension of the method to the nonconvex case. This, however, seems to be a nontrivial task, since underlying the method are properties of the improvement function defined by (1.2), which strongly rely on convexity. But if a suitable extension of Theorem 2.1 to the nonconvex case can be found, then one can try to extend the algorithm by using the subgradient locality measures instead of the linearization errors, along the lines of [28, 16].

Acknowledgment. We thank the two anonymous referees for careful reading of the first version and for constructive suggestions.

REFERENCES

[1] J.F. Bonnans, J.-Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. A family of variable metric proximal point methods. *Mathematical Programming*, 68:15–47, 1995.

[2] J.F. Bonnans, J.-Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization. Theoretical and Practical Aspects.* Universitext. Springer-Verlag, Berlin, 2003.

[3] P.N. Brown. Decay to uniform states in ecological interactions. *SIAM Journal on Applied Mathematics*, 38:22–37, 1980.

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

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

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

[7] R. Fletcher and S. Leyffer. A bundle filter method for nonsmooth nonlinear optimization. Numerical Analysis Report NA/195, Department of Mathematics, The University of Dundee, Scotland, 1999.

[8] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–269, 2002.

[9] A. Frangioni. Solving semidefinite quadratic optimization problems within nonsmooth optimization problems. *Comput. Oper. Res.*, 23(11):1099–1118, 1996.

[10] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.

[11] J.L. Goffin, A. Haurie, and J.Ph. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38(2):284–302, 1992.

[12] M. Hintermüller. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, 20:245–266, 2001.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms.* Number 305-306 in Grund. der Math. Wiss. Springer-Verlag, 1993.

[14] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes.* Lecture Notes in Economics and Mathematical Systems, No. 187, Springer–Verlag, Berlin, Germany, 1981.

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

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

[17] K.C. Kiwiel. An exact penalty function algorithm for nonsmooth convex constrained minimization problems. *IMA Journal of Numerical Analysis*, 5:111–119, 1985.

[18] K.C. Kiwiel. A method for solving certain quadratic programming problems arising in nonsmooth optimization. *IMA Journal of Numerical Analysis*, 6:137–152, 1986.

[19] K.C. Kiwiel. A constraint linearization method for nondifferentiable convex minimization. *Numerische Mathematik*, 51:395–414, 1987.

[20] K.C. Kiwiel. A subgradient selection method for minimizing convex functions subject to linear constraints. *Computing*, 39:293–305, 1987.

[21] K.C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46:105–122, 1990.

[22] K.C. Kiwiel. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Mathematical Programming*, 52:285–302, 1991.

[23] C. Lemaréchal and R. Mifflin. *A set of nonsmooth optimization test problems.* Nonsmooth optimization, C. Lemaréchal and R. Mifflin (eds.), Pergamon Press, Oxford, 1978, pp. 151-165.

[24] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–148, 1995.

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

[26] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained optimization. *Journal of Optimization Theory and Applications*, 102:593–613, 1999.

[27] O.L. Mangasarian. *Nonlinear Programming.* McGraw–Hill, New York, 1969.

[28] R. Mifflin. An algorithm for constrained optimization with semismooth functions. *Mathematics of Operations Research*, 2:191–207, 1977.

[29] R. Mifflin. A modification and extension of Lemarechal's algorithm for nonsmooth minimization. *Mathematical Programming Study*, 17:77–90, 1982.

[30] R. Mifflin. A superlinearly convergent algorithm for one-dimensional constrained minimization problems with convex functions. *Mathematics of Operations Research*, 8:185–195, 1983.

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

[32] O. Pironneau and E. Polak. Rate of convergence of a class of methods of feasible directions. *SIAM Journal on Numerical Analysis*, 10:161–174, 1973.

[33] B. T. Polyak. *Introduction to Optimization.* Optimization Software, Inc., Publications Division, New York, 1987.

[34] P. Rey and C. Sagastizábal. Dynamical adjustment of the prox-parameter in bundle methods. *Optimization*, 51:423–447, 2002.

[35] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2:121–152, 1992.

[36] N. Shor. *Minimization Methods for Non-differentiable Functions.* Springer-Verlag, Berlin, 1985.

[37] M. V. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119:151-165, 2003.

[38] Zoutendijk. *Methods of Feasible Directions.* Elsevier, Amsterdam, 1960.