SIAM J. Optim.
Vol. 20, No. 5, pp. 2442–2473

A REDISTRIBUTED PROXIMAL BUNDLE METHOD FOR NONCONVEX OPTIMIZATION[*]

WARREN HARE[†] and CLAUDIA SAGASTIZÁBAL[‡]

Abstract. Proximal bundle methods have been shown to be highly successful optimization methods for unconstrained convex problems with discontinuous first derivatives. This naturally leads to the question of whether proximal variants of bundle methods can be extended to a nonconvex setting. This work proposes an approach based on generating cutting-planes models, not of the objective function as most bundle methods do but of a local convexification of the objective function. The corresponding convexification parameter is calculated "on the fly" in such a way that the algorithm can inform the user as to what proximal parameters are sufficiently large that the objective function is likely to have well-defined proximal points. This novel approach, shown to be sound from both the objective function and subdifferential modelling perspectives, opens the way to create workable nonconvex algorithms based on nonconvex $\mathcal{VU}$ theory. Both theoretical convergence analysis and some encouraging preliminary numerical experience are provided.

Key words. nonconvex optimization, nonsmooth optimization, proximal point, prox-regular, bundle method, lower-$\mathcal{C}^2$

AMS subject classifications. Primary, 90C26, 49J52; Secondary, 65K10, 49J53, 49M05

DOI. 10.1137/090754595

1. Introduction.

We consider the nonsmooth unconstrained optimization problem

$$\min_{x\in\mathbb{R}^N} f(x).$$

Bundle methods are currently among the most efficient optimization methods in the presence of discontinuous first derivatives, at least for convex objective functions [23]. In this paper we build on previous research on determining proximal points for nonconvex functions in order to develop a basic proximal bundle method that is suitable for nonconvex objective functions.

Initially, bundle methods were based on subdifferential estimates that asymptotically ensure satisfaction of the first order optimality conditions [21], [27], [22], [24], [28], [29]. In these *dual* methods not much attention is paid to the *primal* view of how to model the objective function by using tangent hyperplanes. Primal forms of convex bundle methods, sometimes referred to as stabilized cutting planes or proximal bundle methods, were mostly developed in the 1990s; see [16, Ch. XV] and references therein.

The dual insight was so pervasive that practically all nonconvex bundle algorithms are modifications of some convex forerunner, with a "fixed" model function that is not theoretically supported from the primal point of view. Basically, such fixes

[*]Received by the editors April 1, 2009; accepted for publication (in revised form) April 26, 2010; published electronically June 18, 2010.
http://www.siam.org/journals/siopt/20-5/75459.html

[†]IRMACS, Simon Fraser University, 8888 University Dr., Burnaby, BC, V5A, 1S6, Canada (whare@irmacs.sfu.ca). Research for this author was supported in part by NSERC DG 355571-2008 and the MoCSSy CTEF SFU.

[‡]Cepel-Elebrobras IMPA, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil (sagastiz@impa.br). Research for this author was partially supported by CNPq grant 300345/2008-9, PRONEX-Optimization, NSF grant 0707205, and AFOSR grant FA9550-08-1-0370. This author was on leave from INRIA Rocquencourt, France.

consist of redefining linearization errors to enforce nonnegativity. At a given point, the *linearization error* of a given line tangent to f is the difference between f and the line, evaluated at the point under consideration; see [16, Def. XIV.1.2.7] and (3) below. If f is convex, tangent lines are "cutting planes" supporting the graph of f, and linearization errors are always nonnegative. If linearization errors are nonnegative, model functions, usually defined as the maximum of tangent lines, are lower approximations to the objective function. This feature is crucial to prove convergence of most bundle methods [16, Vol. II], [4], [17]. However, in the nonconvex case, since tangent hyperplanes may not support the graph of the objective function, linearization errors can be negative. The corresponding model function does not stay below f and may even cut off a region containing a minimizer. This drawback was circumvented by forcing the linearization error to remain positive by purely heuristic methods, such as replacing negative values with a quadratic term, or with the absolute value of the linearization error. (See [29] for a general study on this subject.)

In this work we present a new nonconvex bundle method, which is more deeply rooted in a logical primal-dual assessment of the cutting-planes model. The algorithm, of the proximal form, is largely based on the work [12], laying the ground for the algorithm's convergence analysis. The algorithm generates approximate proximal points, computed by using a variation of the algorithm presented in [12], in which (exact) proximal points of a special cutting-planes model are used to compute increasingly accurate approximations to the correct proximal point. The cutting-planes model is special in the sense that it no longer models the objective function f but rather certain local convexification, centered at the current serious step (see (4)). Such local convexification is modified dynamically in order to always yield nonnegative linearization errors and stems from redistributing the proximal parameter of the objective function into two terms; see (6) below.

Like our method, many previous bundle methods for nonconvex objective functions follow a proximal methodology [17], [26], [35], [19], [25], [36], [8], [9], [11]. Also like our method, the majority of these methods recognize the benefits of enforcing positive linearization errors. However, unlike our method, previous methods that desired positive linearization errors did so by redefining them through a variety of heuristics, for example, by applying absolute values to the linearization errors [19], by applying quadratic penalty terms to the linearization errors [25], or by applying some combination of both [29], [17], [26], [35], [11]. (A more recent approach involves splitting linearization errors into two sets of positive and negative errors to create two model functions [8], [9].) Some penalty term approaches fix the penalization parameter a priori (terminating if the penalty appears insufficient); other proposals increase the penalty iteratively by some heuristical measures. However, none of the past approaches examines how the penalty affects the primal objective information. For this reason, the corresponding model functions cannot be clearly related to the original objective function.

In a manner conversely, our approach begins with the objective function and shows that a convexification term can be used that will cause linearization errors to become positive. This maintains a connection between the model functions and the original objective function. By approaching the problem from this direction, we create a better understanding of why the algorithm works and how it might be further employed. For example, the algorithm developed in this paper is not only shown to converge on a variety of nonconvex functions but is also designed to inform the user as to what proximal parameters are sufficiently large that the objective function is likely to have well-defined proximal points (information comparable to having a suitable penalty parameter in previous approaches).

Nonsmooth methods are, in general, linearly convergent at best. This is not only the case for the bundle family but also of the gradient sampling method [3], its lesser-known antecessor [10], and for [32] and the derivative-free methods [1] and [2]. The interest in devising proximal nonconvex bundle methods lies on the "smoothing" effect of the proximal point operator and its potential to speed up the algorithm's convergence. Informing the user of the algorithm's estimate of the proximal threshold for the objective function should allow for fast nonconvex algorithms, based on the nonconvex $\mathcal{VU}$ theory in [30], [31] and the nonconvex partly smooth theory in [13].

The remainder of this paper is organized as follows. In section 2 we review some variational analysis definitions and results required for this work. Section 2 also includes the main assumption for the functions considered in this work, some examples of functions that satisfy the assumption, and some basic results arising from the assumptions. In section 3 we review how bundle methods work in a convex setting and discuss how requirements change in a nonconvex setting. This section states most of our notation. In section 4 we provide the details of the algorithm developed in this paper, followed by results showing that the algorithm is well defined. Section 5 examines the convergence properties of the algorithm. Section 6 contains encouraging results of some preliminary numerical testing for the algorithm. We give some concluding remarks in section 7.

2. Background, assumptions, and notation.

In this section we recall concepts and results of variational analysis that will be of use in this paper. Notably, we make use of the limiting subdifferential, denoted by $\partial f(\bar{x})$ in [34, Def. 8.3, p. 301]. More precisely, having a regular subdifferential of f at $\bar{x}$,

$$\hat{\partial} f(\bar{x}) := \{g \in \mathbb{R}^N : \liminf_{x \to \bar{x}} \inf_{x \neq \bar{x}} \frac{f(x) - f(\bar{x}) - \langle g, x - \bar{x}\rangle}{|x - \bar{x}|} \geq 0\}\,,$$

the limiting subdifferential is defined by

$$\partial f(\bar{x}) := \lim_{x \to \bar{x}} \sup_{f(x) \to f(\bar{x})} \hat{\partial} f(x).$$

We call elements of this subdifferential *subgradients* and generally represent them with the variable g.

We say a function f is *prox-bounded* if there exists $R \geq 0$ such that the function $f + R\frac{1}{2}|\cdot|^2$ is bounded below. The corresponding *threshold* (of prox-boundedness) is the smallest $r_{pb} \geq 0$ such that $f + R\frac{1}{2}|\cdot|^2$ is bounded below for all $R > r_{pb}$.

Our nonconvex proximal bundle method is given for *lower*-C^2 functions that we define following the equivalence shown in [34, Thm. 10.33, p. 450]. Specifically, the function f is lower-$\mathcal{C}^2$ on an open set $\mathcal{O}$ if f is finite valued on $\mathcal{O}$ and for any point x in $\mathcal{O}$ there exists a threshold $r_{lC2}(x) > 0$ such that $f + \frac{r}{2}|\cdot|^2$ is convex on an open neighborhood $\mathcal{O}'$ of x for all $r > r_{lC2}(x)$.

We now state our basic assumption on the objective function f, depending on given parameters x^0 and M_0.

$$\boxed{\begin{array}{c}\text{Given } x^0 \in \mathbb{R}^N \text{ and } M_0 \geq 0 \text{ there exists}\\ \text{an open bounded set } \mathcal{O} \text{ and a function } F \text{ such that}\\ \mathcal{L}_0 := \left\{x \in \mathbb{R}^n : f(x) \leq f(x^0) + M_0\right\} \subset \mathcal{O}, \text{ and}\\ F \text{ is lower-}\mathcal{C}^2 \text{ on } \mathcal{O} \text{ with } F \equiv f \text{ on } \mathcal{L}_0.\end{array}} \tag{1}$$

To stress the dependence of this assumption on the parameters, we refer to it as $(1)_{(x^0,M_0)}$ when appropriate.

Remark 1. Assumption (1) essentially states that the objective function f is lower-$\mathcal{C}^2$ near the minimizer(s) of the problem. Indeed, if f is lower-$\mathcal{C}^2$ on an open bounded set $\mathcal{O}$ satisfying

$$\mathcal{L}_0 := \{x \in \mathbb{R}^n : f(x) \leq f(x^0) + M_0\} \subset \mathcal{O}\,,$$

then f satisfies $(1)_{(x^0,M_0)}$. We state assumption (1) in the more general form above because both x^0 and M_0 are given when initializating Algorithm 1 and they are not calculated values (x^0 plays the role of the first prox-center, while M_0 is an "unacceptable increase" parameter).

Before moving on to some of the properties of functions satisfying assumption $(1)_{(x^0,M_0)}$, we give some examples of such functions. In the first example, we note that Lipschitz functions that are prox-regular (in the sense of Poliquin and Rockafellar [33]) are lower-$\mathcal{C}^2$. Since prox-regularity is essential when working with proximal points in a nonconvex setting ([14]), the example shows that lower-$\mathcal{C}^2$ functions are a useful subset of Lipschitz functions in our setting.

Example 1 (prox-regular Lipschitz). Let f be a prox-regular locally Lipschitz function. Then f is lower-$\mathcal{C}^2$ [34, Prop. 13.33 and Def. 9.1]. In particular, if f is (globally) semismooth (in the sense of Mifflin [27]) and prox-regular, then f is (globally) lower-$\mathcal{C}^2$; see also [15].

In the next example, we describe one manner of ensuring the level sets are bounded, as in assumption (1).

Example 2 (level coercive lower-$\mathcal{C}^2$). Let the function f be a globally lower-$\mathcal{C}^2$. Suppose that f is level coercive (i.e., bounded below on bounded sets and $\liminf_{|x|\to\infty} f(x)/|x| > 0$). Then by [34, Cor. 3.27, p. 92], the level sets of f are bounded. Thus, for any $M_0 > 0$ and any x^0, one has that f satisfies assumption $(1)_{(x^0,M_0)}$. (Here $\mathcal{O}$ can be any bounded set containing $\mathcal{L}_0$.)

Let the function f be a lower-$\mathcal{C}^2$ on the open set $\mathcal{O}$ containing a minimizer, $\bar{x} \in \operatorname{argmin} f$. Suppose that f is level coercive, so the level sets of f are bounded. Then f satisfies assumption $(1)_{(x^0,M_0)}$ for any x^0 and M_0 such that $\mathcal{L}_0 \subset \mathcal{O}$. (Since $\mathcal{L}_0$ is bounded, we may reduce $\mathcal{O}$ to a bounded open set containing $\mathcal{L}_0$.)

The final two examples demonstrate methods of building functions which satisfy assumption $(1)_{(x^0,M_0)}$.

Example 3 (construction from lower-$\mathcal{C}^2$). Let the function f be a lower-$\mathcal{C}^2$ function on the (possibly unbounded) set $\mathcal{O}$. Given $\varepsilon > 0$, $M > 0$, and a point $\bar{y} \in \mathcal{O}$, define $\tilde{f}$ by $\tilde{f}(x) := \max\{f(x), \varepsilon\frac{1}{2}|x-\bar{y}|^2 - M\}$. Then $\tilde{f}$ is lower-$\mathcal{C}^2$ (on $\mathcal{O}$) and level coercive (as $|x-\bar{y}|^2$ is level coercive), so the conclusions of Example 2 hold. (Note that by selecting M sufficiently large, it is easy to ensure that $\tilde{f}(x) = f(x)$ for all x near $\bar{y}$.)

Example 4 (construction via indicator functions). Let the function f be a lower-$\mathcal{C}^2$ function on $\mathcal{O}^*$ (possibly unbounded). Let C be any compact subset of $\mathcal{O}^*$, and define $\tilde{f}$ by $\tilde{f}(x) := f(x) + \iota_C(x)$, where ι is the indicator function of C ($\iota_C(x)$ equals 0 on C and infinity elsewhere). Let $x^0 \in C$ and $M_0 \geq 0$; then $\tilde{f}$ satisfies assumption $(1)_{(x^0,M_0)}$. Indeed, in this case the function "F" is the original function f, while the open bounded set "$\mathcal{O}$" is any open subset of $\mathcal{O}^*$ containing C. (Note that $\mathcal{L}_0 \subseteq C$, as $\iota_C(x) = \infty$ for $x \notin C$.)

The following result gathers some important consequences of our assumption, essentially related to the existence of uniform bounds for the various thresholds involved (lower-C^2, Lipschitz continuity, etc.).

PROPOSITION 1. *For a function f satisfying* $(1)_{(x^0,M_0)}$, *the following holds:*

(a) *The level set $\mathcal{L}_0$ is nonempty and compact.*

(b) *The function f is bounded below and prox-bounded with threshold $r_{pb} = 0$.*

(c) *There exists $\rho^{id} > 0$ such that, for any $\rho \geq \rho^{id}$ and given any $y \in \mathcal{L}_0$,*

$$\text{the function } f + \frac{\rho}{2}|\cdot - y|^2 \text{ is convex on } \mathcal{L}_0.$$

(d) *The function f is Lipschitz continuous on $\mathcal{L}_0$.*

Proof. Let F be the lower-$\mathcal{C}^2$ function from assumption (1), coinciding with f on $\mathcal{O}$. Being lower-C^2, the function F is continuous and finite valued on the open set $\mathcal{O}$, which contains the level set $\mathcal{L}_0$. Thus $\mathcal{L}_0$ is closed. By assumption (1), the level set $\mathcal{L}_0$ is bounded and therefore compact. As $x^0 \in \mathcal{L}_0$, we see $\mathcal{L}_0$ is nonempty, as claimed in item (a).

Since the remaining results are concerned only with the behavior of f on $\mathcal{L}_0$ and $\mathcal{L}_0 \subset \mathcal{O}$, where f and F agree, we may assume without loss of generality that f and F are the same function.

Since f is finite and continuous on the compact level set $\mathcal{L}_0$, f is bounded below on this set. By the definition of $\mathcal{L}_0$, this implies f is bounded below. Any function which is bounded below is prox-bounded with threshold $r_{pb} = 0$.

By [34, Prop. 10.54], there exists an open set $\mathcal{O}'$ satisfying $\mathcal{L}_0 \subset \mathcal{O}' \subseteq \mathcal{O}$ and $\rho^{id} > 0$ such that for any point $y \in \mathcal{O}'$, the function $f + \rho\frac{1}{2}|\cdot - y|^2$ is convex on $\mathcal{O}'$ (and therefore on $\mathcal{L}_0$) for any $\rho \geq \rho^{id}$.

All lower-$\mathcal{C}^2$ functions are locally Lipschitz continuous [34, Thm. 10.31]. The compactness of $\mathcal{L}_0$ allows one to find a Lipschitz constant that holds for all of $\mathcal{L}_0$. □

Another important consequence of our assumption $(1)_{(x^0,M_0)}$ is that the proximal point mapping p_R, defined as

$$p_R f(x) := \operatorname*{argmin}_y \left\{ f(y) + R\frac{1}{2}|x - y|^2 \right\},$$

is single-valued and Lipschitz continuous on $\mathcal{L}_0$, provided the prox-parameter R is sufficiently large. Furthermore, stationary points of f are characterized as fixed points of the proximal point mapping:

$$\bar{x} \in \mathcal{L}_0 \text{ is a stationary point of } f \text{ if and only if } \bar{x} = p_R f(\bar{x}),$$

again provided R is sufficiently large. By examining [34, Thm 2.26] it is clear that, in this case, R sufficiently large means that

$$R > \rho^{id}, \tag{2}$$

where ρ^{id} is the value in item (c) in Proposition 1. To see this, note that, by the definition of $\mathcal{L}_0$, for any $x \in \mathcal{L}_0$ one must have $\operatorname{argmin}_y\{f(y) + R\frac{1}{2}|x - y|^2\} = \operatorname{argmin}_y\{f(y) + R\frac{1}{2}|x - y|^2 + \iota_{\mathcal{L}_0}\}$ (if $y \notin \mathcal{L}_0$, then $f(x) < f(y) + R\frac{1}{2}|x - y|^2$, so $y \notin \operatorname{argmin}_y\{f(y) + R\frac{1}{2}|x - y|^2\}$).

The characterization of stationary points as fixed points of the proximal point mapping, together with the local convexity property in item (c) in Proposition 1, plays a fundamental role in our development. Suppose, for the moment, we are in an ideal situation, where the convexity and proximal threshold ρ^{id} is known. Then, given any $R > \rho^{id}$, we could apply a convex bundle algorithm for iteratively computing a fixed point of $p_R f$ by exploiting the relation

$$p_R f(x) = p_{R-\rho^{id}}\Big(f + \frac{\rho^{id}}{2}|\cdot - x|^2\Big)(x).$$

The redistributed proximal bundle algorithm given in section 4 below is nothing but an implementable form of such an ideal algorithm. More precisely, the minimum ideal threshold ρ^{id} is estimated along iterations, using data generated by the algorithm, accumulated in a *bundle* of information.

3. Bundling in a nonconvex setting.

Suppose, for the moment, f is convex. At any iteration n, classical bundle methods keep memory of the iterative process in a bundle of information, essentially collecting past function and subgradient values

$$\bigcup_{i \in I_n} \{(x^i, f_i = f(x^i), g^i \in \partial f(x^i))\},$$

where I_n denotes an index set of previous iterations, i.e., $I_n \subseteq \{0, 1, \dots, n\}$. These methods also keep track of $f(\hat{x}^{k(n)})$, the "best" function value obtained until iteration n, evaluated at the "serious" step k, corresponding to some past iterate i_k. (In the future, when it is clear from the context, we drop the explicit dependence of k on the current iteration index n to alleviate notation.)

The subsequence of serious points has decreasing objective values, and, under reasonable assumptions, its cluster points minimize the function. Therefore, it is sometimes convenient to write the bundle information by referring it to the current serious step. For a convex function f, the rewriting involves determining the linearization errors for f (at $\hat{x}^k$), defined by

$$e_i^k = f(\hat{x}^k) - (f_i + \langle g^i, \hat{x}^k - x^i\rangle). \tag{3}$$

The reformulated bundle data now consist of the current serious step

$$\{\hat{x}^k, f_k = f(\hat{x}^k), g^k \in \partial f(\hat{x}^k)\}$$

and the approximate subgradients,

$$\bigcup_{i \in I_n} \{(e_i^k, g^i \in \partial_{e_i^k} f(x^k))\},$$

where $\partial_e f$ is the e-subdifferential in convex analysis. An additional advantage of using this reformulated bundle is that it opens the way of the mechanism known as *bundle compression*, that allows to keep bounded the cardinality of the index set I_n as $n \to \infty$, without impairing convergence of the method.

In order to define the bundle of information in our nonconvex setting, recall first that for a convex function f, linearization errors are always nonnegative. A nonconvex function f may, by contrast, yield negative linearization errors that have to be dealt with adequately along the iterative process. For this reason, the approach introduced in [12], and extended here, works with augmented functions

$$f_{\eta_n}^{\hat{x}^k} := f + \eta_n|\cdot - \hat{x}^k|^2/2. \tag{4}$$

Accordingly, we consider an augmented bundle of information:

$$\bigcup_{i\in I_n}\{(e_i^k, d_i^k, \Delta_i^k, g^i)\}, \text{ where } \begin{cases} e_i^k \text{ was defined in (3)},\\ d_i^k = |x^i - \hat{x}^k|^2/2,\\ g^i \in \partial f(x^i),\\ \Delta_i^k = x^i - \hat{x}^k. \end{cases} \tag{5}$$

Note that since the linearization errors and the difference norms and vectors above depend on $\hat{x}^k$, they need to be updated every time the serious step changes. Although at first glance, keeping in the bundle the difference norms d_i^k in addition to the difference vectors Δ_i^k may seem superfluous, this is not the case when introducing the mechanism of compression. We explain below how to handle the compression of bundle elements in a nonconvex setting; see (9).

As usual with proximal variants of bundle methods, the serious point $\hat{x}^k$ represents the prox-center for the current iteration. The algorithm proceeds by defining "candidate" points x^{n+1} as the solution to a certain quadratic programming (QP) problem. The next serious point $\hat{x}^{k+1}$ will be a candidate point satisfying the serious step condition given in Step 3 below. We shall always assume an initial point $x^0 (= \hat{x}^0)$ is given.

For convenience, we drop, for the moment, iteration indices in our notation. To define the QP problem, the current prox-parameter R is split into two nonnegative terms η and μ satisfying $R = \eta + \mu$. These terms play two distinct roles, derived from the relation

$$p_R f(\hat{x}) = p_\mu(f_\eta^{\hat{x}})(\hat{x}), \tag{6}$$

where $f_\eta^{\hat{x}} = f + \eta|\cdot - \hat{x}|^2/2$ as in (4). In this expression, η defines the augmented "convexified" function $f_\eta^{\hat{x}}$, to be modeled by a simple function φ. The remaining parameter, μ, is used as a prox-parameter for the model function. We refer to η and μ as the *convexification parameter* and *model prox-parameter*, respectively. Along the iterative process, R, η, μ, and φ have to be suitably modified. The bundle of past information is used to define a cutting-planes model of the function $f_\eta^{\hat{x}}$:

$$\varphi(y) := \max_i \left\{ \left(f_i + \eta\frac{1}{2}|x^i - \hat{x}|^2 \right) + \langle (g^i + \eta(x^i - \hat{x})), y - x^i \rangle \right\}.$$

An equivalent expression, based on (3) and (5) and written with all the iteration indices, is the following:

$$\varphi_n(y) = f(\hat{x}^k) + \max_{i\in I_n}\{-(e_i^k + \eta_n d_i^k) + \langle (g^i + \eta_n \Delta_i^k), y - \hat{x}^k \rangle\}. \tag{7}$$

By letting $\mathcal{S}_n$ denote the unit simplex in $\mathbb{R}^{|I_n|}$, the candidate point is $x^{n+1} := p_{\mu_n}\varphi_n(\hat{x}^k)$, i.e., the unique solution to a QP problem. The corresponding optimality condition is as follows:

$$\text{there exists } \alpha^n \in \mathcal{S}_n \text{ such that} \\ \begin{cases} x^{n+1} = \hat{x}^k - \dfrac{1}{\mu_n} g_{\eta_n}^{-n} \text{ with} \\ g_{\eta_n}^{-n} := \displaystyle\sum_{i\in I_n} \alpha_i^n (g^i + \eta_n \Delta_i^k) \in \partial\varphi_n(x^{n+1}). \end{cases} \tag{8}$$

In the sequel, we shall use J_n^{act} to denote the set of all "strongly active" subgradients. That is,

$$J_n^{act} := \{i \in I_n : \alpha_i > 0 \text{ in } (8)\}.$$

In view of condition (8), we call the augmented subgradient $g_{\eta_n}^{-n}$ above the *aggregate subgradient*. The corresponding aggregate bundle element is the quadruplet

$$(e^k_{-n}, d^k_{-n}, \Delta^k_{-n}, g^{-n}) := \sum_{i\in I_n} \alpha^n_i (e^k_i, d^k_i, \Delta^k_i, g^i) = \sum_{\ell\in J_n^{act}} \alpha^n_\ell (e^k_\ell, d^k_\ell, \Delta^k_\ell, g^\ell). \tag{9}$$

(We follow the notation in [20], reserving negative indices in I_n for aggregate bundle elements; so, in general, $I_n \subseteq \{-n, -n+1, \dots, 0, 1, \dots, n-1, n\}$.)

Note that

$$\left.\begin{array}{c}\text{for all } \ell \in J^{act} \\ \text{and } \ell = -n\end{array}\right\} \varphi_n(x^{n+1}) = f(\hat{x}^k) - e^k_\ell - \eta_n d^k_\ell + \left\langle g^\ell + \eta_n \Delta^k_\ell, x^{n+1} - \hat{x}^k \right\rangle , \tag{10}$$

where the equality for $\ell \in J_n^{act}$ follows by complementarity, while for $\ell = -n$, it follows from the result for $\ell \in J_n^{act}$, making the convex sum and recalling (9).

Using the aggregate bundle element, from (8) we have

$$g_{\eta_n}^{-n} = g^{-n} + \eta_n \Delta^k_{-n} = \mu_n(\hat{x}^k - x^{n+1}), \tag{11}$$

and the cutting plane $-e^k_{-n} - \eta_n d^k_{-n} + \langle g_{\eta_n}^{-n}, y - x^k \rangle$ is included in the generation of the model function. In general, bundle elements defining the model φ_n may have the form (5) or (9). For the quadruplets from (5), g^i is a genuine subgradient for the function f: $g^i \in \partial f(x^i)$, a relation that fails to hold for the aggregate gradient from (9). Accordingly, we sometimes distinguish between *oracle* and *aggregate* bundle elements. Note that, in both cases, the difference norm vectors are nonnegative: $d^k_i \geq 0$ for all $i \in I_n$. Also, for both oracle and aggregate quadruplets, every time there is a new serious point, the index k changes to $k+1$, and the first three elements in the quadruplet are updated according to the formulæ:

$$\begin{array}{rcl} e_i^{k+1} & = & e_i^k + f(\hat{x}^{k+1}) - f(\hat{x}^k) - \langle g^i, \hat{x}^{k+1} - \hat{x}^k \rangle \\ d_i^{k+1} & = & d_i^k + |\hat{x}^{k+1} - \hat{x}^k|^2/2 - \langle \Delta_i^k, \hat{x}^{k+1} - \hat{x}^k \rangle \\ \Delta_i^{k+1} & = & \Delta_i^k + \hat{x}^k - \hat{x}^{k+1} \end{array} \tag{12}$$

for all $i \in I_n$.

Recall that, in classical bundle methods for convex functions, the bundle consists of the pair (e^k_i, g^i) for which the relation

$$g^i \in \partial_{e^k_i} f(\hat{x}^k)$$

holds. In our method, this pair is replaced with a quadruplet $(e^k_i, d^k_i, \Delta^k_i, g^i)$ for which the relation

$$g^i + \eta_n \Delta^k_i \in \partial_{e^k_i + \eta_n d^k_i} \varphi_n(\hat{x}^k) \quad \text{whenever } e^k_i + \eta_n d^k_i \geq 0 \tag{13}$$

holds. The challenge is therefore to select η_n sufficiently large that $e^k_i + \eta_n d^k_i \geq 0$ for all $i \in I_n$ but sufficiently small to remain manageable. The basis of our method is to make the parameter η_n asymptotically estimate the ideal convexity threshold ρ^{id}. As a result, the model φ_n eventually becomes a lower approximation to a locally convex function $f^{\bar{x}}_{\rho^{id}}$, with nonnegative augmented linearization errors (where the stationary point $\bar{x}$ is a cluster point for the serious step sequence $\{\hat{x}^k\}$).

A first possibility to set the convexification parameter η_n is to define it as the minimal value that keeps the augmented linearization errors nonnegative:

$$\eta_n^{\min} := \max_{\substack{i \in I_n \\ d_i^{k(n)} > 0}} -\frac{e_i^{k(n)}}{d_i^{k(n)}}. \tag{14}$$

(Clearly, $e_i^k + \eta d_i^k \geq 0$ for all $i \in I_n$ whenever $\eta \geq \eta_n^{\min}$.)

The work [12] uses instead the lower bound

$$\tilde{\eta}_n := \max_{\substack{i, j \in I_n \\ i \neq j}} \frac{e_j^k - e_i^k - \langle g^j - g^i, \Delta_j^k \rangle}{d_j^k + d_i^k - \langle \Delta_i^k, \Delta_j^k \rangle},$$

which is enough to ensure that the proximal points of φ_n converge to the proximal point of f. Let i_k be the (past) iterate giving the current serious point: $\hat{x}^k = x^{i_k}$. The following holds:

$$i_k \in I_n \Leftrightarrow (e^k_{i_k}, d^k_{i_k}, \Delta^k_{i_k}, g^{i_k}) = (0, 0, 0, g^{i_k}) \text{ is in the bundle.} \tag{15}$$

In this case, since setting $j = i_k$ in the right-hand-side term defining $\tilde{\eta}_n$ yields (14), we see that the lower bound $\tilde{\eta}_n$ is at least as large as $\eta_n^{\min}$.

Relation (14) guides the choice of the convexification parameter in our algorithm. In addition to helping find the ideal proximal threshold $R > \rho^{id}$ as in (2), the parameter $R_n (= \eta_n + \mu_n)$ is increased when needed along iterations.

4. Algorithmic development.

As discussed in section 1, previous methods for extending bundle methods to a nonconvex setting generally hinge around redefining the linearization errors in order to force them to be positive.

Along these lines, the most common methods stem from the research of [29] and [17], which redefine the linearization errors as the minimum of the absolute value of the linearization error and a small quadratic penalty term (based on the distance from the most recent serious point). Similar methods are employed in [26], [35], [11], while in [25] it is shown that the method can converge without the use of the absolute value function (i.e., replacing linearization errors with the minimum of the linearization error and a small quadratic term). Some approaches that use penalty terms to generate linearization fix the penalty a priori (terminating if the penalty appears insufficient); others alter the penalty dynamically using heuristical measures. Although provably convergent, these methods have the undesirable consequence of redefining linearization errors that are equal to zero. In [19] a method is derived which does not employ a quadratic penalty; instead, it replaces all linearization errors with their absolute value. Also provably convergent, it is unclear how the model functions resulting from such techniques relate to the original objective function.

Our method is most similar to the quadratic penalty method. However, instead of replacing some linearization errors with small quadratic terms, quadratic terms are added to all linearization errors equally. By doing this, we are able to keep track of the relationship between the original objective function and the generated piecewise linear model function. This maintains a powerful connection between the model functions and the original objective function. Our coefficient η_n will be calculated dynamically during each iteration in a manner which forces the linearization errors of a *penalization* of the objective function to be positive. As a result, while previous methods shift the f-hyperplanes defining the cutting-planes model, we shift and tilt such hyperplanes; since we model (4), not only intercepts but also slopes are changed.

In the next subsection we provide pseudocode for the *redistributed bundle algorithm.*

4.1. Redistributed bundle algorithm.

An oracle computing the function value $f(x)$ and one subgradient in $\partial f(x)$ for any $x \in \mathbb{R}^N$ is assumed to be given.

ALGORITHM 1 (REDISTRIBUTED BUNDLE).

Step 0 **(Input and Initialization)**

Select initial starting point $\hat{x}^0$ and an unacceptable increase parameter $M_0 > 0$, a parameter $R_0 > 0$, a stopping tolerance $\text{TOL}_{\text{stop}} \geq 0$, an Armijo-like parameter $m \in (0,1)$, and a convexification growth parameter $\Gamma > 1$. Initialize the iteration counter $n = 0$, the serious step counter $k = k(n) = 0$ with $i_0 = 0$, the bundle index set $I_0 := \{0\}$, and the first candidate point $x^0 := \hat{x}^0$.

Compute the oracle values $f_0 = f(\hat{x}^0)$ and $g^0 \in \partial f(\hat{x}^0)$ and the additional bundle information $(e_0^0, d_0^0, \Delta_0^0) := (0,0,0) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^N$.

Choose the starting prox-parameter distribution $(\mu_0, \eta_0) := (R_0, 0)$.

Step 1 **(Model Generation and QP Subproblem)**

Having the current serious step prox-center $\hat{x}^k$, the bundle $\left\{\left(e_i^k, d_i^k, \Delta_i^k, g^i\right)\right\}_{i \in I_n}$, and the prox-parameter distribution (μ_n, η_n), with $\eta_n \leq R_n$ and $\mu_n = R_n - \eta_n$, define the convex piecewise linear model function φ_n from (7).

Compute

$$x^{n+1} := p_{\mu_n} \varphi_n(\hat{x}^k),$$

with optimal simplicial multipliers α^n from (8), so that the aggregate quadruplet from (9) is available.

Define the predicted decrease

$$\delta_{n+1} := f(\hat{x}^k) + \frac{\eta_n}{2}|x^{n+1} - \hat{x}^k|^2 - \varphi_n(x^{n+1}).$$

Step 2 **(Stopping the Test and New Bundle Information)**

Call the oracle to obtain $f(x^{n+1})$ and $g^{n+1} \in \partial f(x^{n+1})$.

If $\delta_{n+1} \leq \text{TOL}_{\text{stop}}$, then stop with the message

"Algorithm successfully terminated at x^{n+1}."

Otherwise, compute the additional elements defining the new bundle quadruplet:

$$\begin{array}{rcl} \Delta_{n+1}^k & := & x^{n+1} - \hat{x}^k, \qquad d_{n+1}^k := |\Delta_{n+1}^k|^2/2, \text{ and} \\ e_{n+1}^k & := & f(\hat{x}^k) - (f(x^{n+1}) + \langle g^{n+1}, \Delta_{n+1}^k \rangle). \end{array}$$

Select a new index set satisfying

$$I_{n+1} \supseteq \{n+1, i_k\} \quad \text{and} \quad \left\{ \begin{array}{ll} \text{either} & I_{n+1} \supseteq J_n^{act} := \{i \in I_n : \alpha_i^n > 0\} \\ \text{or} & I_{n+1} \supseteq \{-n\}. \end{array} \right.$$

Step 3 **(Serious Step Test)**

Check the descent condition $\qquad f(x^{n+1}) \leq f(\hat{x}^k) - m\delta_{n+1}.$

If this condition is true, declare a serious step:
set $k(n+1) = k+1$, $i_{k+1} = n+1$, $\hat{x}^{k+1} = x^{n+1}$,
update bundle elements according to formulæ (12).
Otherwise, declare a null step:
set $k(n+1) = k(n)$.

Step 4 (Update η)
Apply the rule

$$\begin{cases} \eta_{n+1} := \eta_n & \text{if } \eta_{n+1}^{\min} \leq \eta_n, \\ \eta_{n+1} := \Gamma \eta_{n+1}^{\min} \qquad \text{and } R_n := \mu_n + \eta_{n+1} & \text{if } \eta_{n+1}^{\min} > \eta_n, \end{cases} \tag{16}$$

where $\eta_{n+1}^{\min}$ is given by (14), written with n replaced by $n+1$.

Step 5 (Update μ)
If $f(x^{n+1}) > f(\hat{x}^k) + M_0$, then the objective increase is unacceptable;
restart the algorithm by setting

$$\begin{aligned} &\eta_0 := \eta_n,\ \mu_0 := \Gamma\mu_n,\ R_0 := \eta_0 + \mu_0, \\ &x^0 := \hat{x}^k,\ (e_0^0, d_0^0, \Delta_0^0) := (0,0,0), \\ &k(0) := 0,\ i_0 = 0,\ I_0 := \{0\}, \\ &n := 0, \end{aligned}$$

and loop to Step 1.
Otherwise, in the case of the serious step, increase k by 1.
In all cases, increase n by 1, and loop to Step 1. □

In Step 1, the dual problem of the QP defining x^{n+1} consists of minimizing a quadratic function over a simplicial set. For this type of structured QP, active set methods such as [18] or [7] are recommended. The update for the model convexification parameters in Step 4 is done to ensure $\eta_n \geq \eta_n^{\min}$ for all iterations, so that $e_{-n}^k + \eta_n d_{-n}^k \geq 0$. Therefore, the predicted decrease defined in Step 1, rewritten in the form

$$\delta_{n+1} = \frac{R_n + \mu_n}{2} |x^{n+1} - \hat{x}^k|^2 + e_{-n}^k + \eta_n d_{-n}^k, \tag{17}$$

obtained by combining (10) for $\ell = -n$ and (11), is also nonnegative.

The potential reset in Step 5 ensures that eventually all bundle points will be in the set $\mathcal{L}_0$. In Lemma 1, we show that there is only a finite number of such restarts.

Lemma 1 (Algorithm 1 is well defined). *Consider the sequence of iterates $\{x^n\}$ generated by Algorithm 1. If the function f satisfies assumption $(1)_{(x^0,M_0)}$ and $I_n \supset \{i_k\}$, there can be only a finite number of restarts in Step 5. Hence, eventually the sequence $\{x^n\}$ lies entirely in $\mathcal{L}_0$, and the model prox-parameter sequence $\{\mu_n\}$ becomes constant.*

Proof. Iterates x^{n+1} are always well defined because the model functions φ_n are convex.

To see there is a finite number of restarts in Step 5, we first note that by assumption $(1)_{(x^0,M_0)}$, the function f is Lipschitz continuous on $\mathcal{L}_0$ (Prop. 1(d)). Let the Lipschitz constant of f on $\mathcal{L}_0$ be L. By the Lipschitz continuity of f, there exists $\varepsilon > 0$ such that for any $\bar{x} \in \{x : f(x) \leq f(x^0)\}$, the open ball $B_\varepsilon(\bar{x})$ is contained within $\mathcal{L}_0$. (Indeed, $\varepsilon = M_0/L$ suffices.)

Next note that

$$\begin{aligned} p_{\mu_n}\varphi_n(\hat{x}^k) &= \operatorname{argmin}_y \{\varphi_n(y) + \tfrac{\mu_n}{2}|y - \hat{x}^k|^2\} \\ &\in \{y : \varphi_n(y) + \tfrac{\mu_n}{2}|y - \hat{x}^k|^2 \leq \varphi_n(\hat{x}^k) + \tfrac{\mu_n}{2}|\hat{x}^k - \hat{x}^k|^2\}. \end{aligned}$$

Since $i_k \in I_n$, the inclusion in (13) written for $i = i_k$ is $g^{i_k} \in \partial\varphi_n(\hat{x}^k)$ by (15). Since it also holds that $g^{i_k} \in \partial f(\hat{x}^k)$, we have that $|g^{i_k}| \le L$ and, hence,

$$\begin{array}{rcl} p_{\mu_n}\varphi_n(\hat{x}^k) & \in & \{y : \varphi_n(\hat{x}^k) + \langle g^{i_k}, y - \hat{x}^k\rangle + \frac{\mu_n}{2}|y - \hat{x}^k|^2 \le \varphi_n(\hat{x}^k)\} \\ & \in & \{y : -|g^{i_k}||y - \hat{x}^k| + \frac{\mu_n}{2}|y - \hat{x}^k|^2 \le 0\} \\ & \in & \{y : \frac{\mu_n}{2}|y - \hat{x}^k|^2 \le L|y - \hat{x}^k|\} \\ & \in & \{y : |y - \hat{x}^k| \le \frac{2L}{\mu_n}\}. \end{array}$$

As μ_n increases during each Step 5 restart, eventually μ_n will become large enough that $2L/\mu_n < \varepsilon$. Noting that $f(\hat{x}^k) < f(x^0)$ for any new $\hat{x}^k$ generated in the algorithm (see Step 3) completes the proof. □

5. Convergence theory.

We now examine the convergence properties of the algorithm. To prove convergence of the case when there is a last serious step followed by infinitely many null steps, we begin by showing that the model functions employed within the algorithm satisfy the conditions used in [12].

5.1. Model properties.

The family of model functions $\{\varphi_n\}$ differs from the one employed in [12] in two important points. First, the rule applied in (16) uses the value $\eta^{\min}$ as a switch (instead of using the bigger value $\tilde{\eta}$). Second, it is possible to replace active bundle elements by the aggregate information (9). (The former variant [12] does not allow to erase active bundle elements.)

Our next goal will be to show that conditions (3) and (6) from [12], crucial for convergence, remain valid for the modified model functions.

More precisely, condition (3) in [12] states that the following five subconditions hold:

(18a) φ_n is a convex function,

(18b) $\varphi_n(\hat{x}^k) \le f(\hat{x}^k)$ for all $w \in \mathbb{R}^N$,

(18c) $\varphi_{n+1}(w) \ge \varphi_n(x^{n+1}) + \mu_n \left\langle \hat{x}^k - x^{n+1}, w - x^{n+1}\right\rangle$ if x^{n+1} is a null step,

(18d) $\varphi_n(w) \ge f(x^n) + \eta_n d_n^k + \langle g^n + \eta_n \Delta_n^k, w - x^n\rangle$ for some $g^n \in \partial f(x^n)$,

(18e) $\underline{\mu} = \mu_n$ and $\eta_n = \overline{\eta}$ for some positive $\underline{\mu}$, nonnegative $\overline{\eta}$, and n sufficiently large.

As for condition (6) in [12], it states that

(19)
$$\varphi_n(w) \le f(w) + \overline{\eta}\frac{1}{2}|w - \hat{x}^{k(n)}|^2 \quad \text{for all } w \text{ near any accumulation point of } \{x^n\}.$$

LEMMA 2 (Conditions (18a–18d); (3a–3d) in [12]). *Consider the family of model functions given by* (7) *and defined by Algorithm* 1. *The following holds:*

(a) *Condition* (18a) *is always satisfied.*

(b) *If $\eta_n \ge \eta_n^{\min}$ for $\eta_n^{\min}$ defined in* (14), *then condition* (18b) *is satisfied.*

(c) *If $\eta_{n+1} = \eta_n$ and either $I_{n+1} \supset J_n^{act}$ or $I_{n+1} \supset \{-n\}$, then condition* (18c) *is satisfied.*

(d) *If $I_n \supset \{n\}$, then condition* (18d) *is satisfied.*

Proof. The first assertion is clear since model functions are generated as the maximum of affine functions.

Item (b) follows from the inequality for η_n, recalling that $\varphi_n(\hat{x}^k) = f(\hat{x}^k) + \max_{i\in I_n}\{-(e_i^k + \eta_n d_i^k)\}$ and that (14) ensures $e_i^k + \eta_n d_i^k \ge 0$.

To see item (c), suppose that x^{n+1} is a null step and $\eta_{n+1} = \eta_n$. Since x^{n+1} is a null step, we have $k(n+1) = k(n) = k$. By definition (7) of φ_{n+1}, using $\eta_{n+1} = \eta_n$, for all $w \in \mathbb{R}^N$ and all $\ell \in I_{n+1}$ we know that

$$\varphi_{n+1}(w) \geq f(\hat{x}^k) - e_\ell^k - \eta_n d_\ell^k + \langle g^\ell + \eta_n \Delta_\ell^k, w - \hat{x}^k \rangle. \tag{20}$$

In particular, (20) holds for all $\ell \in J_n^{act}$ or $\ell = -n$ in the index set I_{n+1} by assumption. For such indices, (10) implies that

$$f(\hat{x}^k) - e_\ell^k - \eta_n d_\ell^k = \varphi_{n+1}(x^{n+1}) - \langle g^\ell + \eta_n \Delta_\ell^k, x^{n+1} - \hat{x}^k \rangle.$$

With this relation, we obtain in (20), for all $\ell \in J_n^{act}$ or $\ell = -n$,

$$\begin{array}{rcl} \varphi_{n+1}(w) & \geq & \varphi_{n+1}(x^{n+1}) + \langle g^\ell + \eta_n \Delta_\ell^k, w - \hat{x}^k + \hat{x}^k - x^{n+1} \rangle \\ & = & \varphi_{n+1}(x^{n+1}) + \langle g^\ell + \eta_n \Delta_\ell^k, w - x^{n+1} \rangle. \end{array}$$

For the case when $I_{n+1} \supset \{-n\}$, since $g^{-n} + \eta_n \Delta_{-n}^k = \hat{x}^k - x^{n+1}$ by (11), the relation above for $\ell = -n$ is just item (c). As for the case of $I_{n+1} \supset J_n^{act}$, we can sum the above inequality by using the convex multipliers α_ℓ^n and recall (9) and (11) to obtain the desired result.

Item (d) is straightforward from the definition of φ_n, recalling that the bundle quadruplet with index n is always an oracle one, given by (5). ▯

5.2. Asymptotic behavior of Algorithm 1. We have seen from Lemma 2 that in Algorithm 1, the choice of index sets I_{n+1} in Step 2 and the update (16) in Step 4 ensure satisfaction of conditions (18a), (18b), and (18d) at every iteration. By contrast, condition (18c) is satisfied only eventually, once the convexification parameters stabilize (i.e., once (18e) holds). We next show that the convexification parameter must eventually stabilize.

LEMMA 3 (eventual stabilization of parameters). *Consider the family of model functions given by* (7) *and defined by Algorithm* 1. *If the function f satisfies assumption* $(1)_{(x^0, M_0)}$, *there exists an iteration $n' > 0$ such that all the parameter sequences stabilize:*

$$\eta_n = \bar{\eta}\,, \mu_n = \bar{\mu}\,, \text{ and } R_n = \bar{R} := \bar{\mu} + \bar{\eta} \text{ for all } n \geq n'.$$

As a result, condition (18) *(i.e.,* (3) *in* [12]*) is eventually satisfied.*

If, in addition, $\bar{\eta} \geq \rho^{id}$, then

$$\varphi_n(w) \leq f(w) + \bar{\eta} |w - \hat{x}^{k(n)}|^2/2 \text{ for all } w \in \mathcal{L}_0 \text{ and for all } n \geq n',$$

and condition (19) *(i.e.,* (6) *in* [12]*) holds.*

Proof. By Lemma 1, there is a finite number of restarts in Step 5 of Algorithm 1. Once there are no more restarts, $\mu_n = \bar{\mu}$ and the update of the convexification parameter in Step 4 is nondecreasing; in (16), either $\eta_{n+1} = \eta_n$ or $\eta_{n+1} = \Gamma \eta_{n+1}^{\min} > \Gamma \eta_n$ with $\Gamma > 1$. For the sequence $\{\eta_n\}$ not to stabilize at some value $\bar{\eta}$, there must be an infinite subsequence of iterations at which the convexification parameter is increased by a factor of at least Γ. But this leads to a contradiction since in this case (Prop. 1(c)), for some iteration n_c, the function $f + \eta_{n_c} |\cdot - \hat{x}^{k(n_c)}|^2/2$ is convex on $\mathcal{L}_0$. For this particular iteration, one will have $e_i^k + \eta_{n_c} d_i^k \geq 0$ for all $i \in I_{n_c}$ (the linearization error for a cutting-planes model of a convex function is always nonnegative). Hence,

$$\eta_{n_c} \geq \max_{i \in I_{n_c}} -\frac{e_i^k}{d_i^k} = \eta_{n_c+1}^{\min},$$

and therefore, from that iteration onward, the update (16) will leave unchanged the convexification parameter: $\eta_{n_c+j} = \eta_{n_c}$ for all $j \geq 0$. The desired result follows from Lemma 2.

The final assertion follows from noting that when $\bar\eta \geq \rho^{id}$, the augmented function $f_{\bar\eta}^{\hat{x}^{k(n)}} = f + \bar\eta|\cdot - x^{k(n)}|^2/2$ is convex on the level set $\mathcal{L}_0$ (Prop. 1(c)), and, hence, the model φ_n remains below the augmented function. □

In order to examine the convergence properties of Algorithm 1, we set $\mathrm{TOL_{stop}} = 0$. Note that if the algorithm stops at some iteration n with $\delta_{n+1} = 0$, by (17) this means that

$$f(\hat{x}^k) + \eta_n \frac{1}{2}|x^{n+1} - \hat{x}^k|^2 = \varphi_n(x^{n+1})$$

Applying that $f(\hat{x}^k) = \varphi_n(\hat{x}^k)$ and $\varphi_n(x^{n+1}) = p_{\mu_n}\varphi_n(\hat{x}^k)$, we see

$$\begin{aligned} \varphi_n(x^{n+1}) + \mu_n \tfrac{1}{2}|x^{n+1} - \hat{x}^k|^2 &\leq \varphi_n(\hat{x}^k) \\ f(\hat{x}^k) + (\mu_n + \eta_n)\tfrac{1}{2}|x^{n+1} - \hat{x}^k|^2 &\leq \varphi_n(\hat{x}^k) \\ f(\hat{x}^k) + R_n \tfrac{1}{2}|x^{n+1} - \hat{x}^k|^2 &\leq f(\hat{x}^k), \end{aligned}$$

which shows that $x^{n+1} = \hat{x}^k$.

Therefore, $\hat{x}^k = p_{\mu_n}\varphi_n(\hat{x}^k)$. However, suppose η_n is sufficiently large for $f + \eta_n|\cdot - \hat{x}^k|^2$ to be convex on $\mathcal{L}_0$. This would imply that

$$\begin{aligned} f(\hat{x}^k) = \varphi_n(\hat{x}^k) &\leq \varphi_n(w) + \mu_n \tfrac{1}{2}|w - \hat{x}^k|^2 && \text{for all } w \in \mathbb{R}^n \\ &\leq f(w) + \eta_n \tfrac{1}{2}|w - \hat{x}^k|^2 + \mu_n \tfrac{1}{2}|w - \hat{x}^k|^2 && \text{for all } w \in \mathcal{L}_0 \\ &\leq f(w) + R_n \tfrac{1}{2}|w - \hat{x}^k|^2 && \text{for all } w \in \mathbb{R}^n. \end{aligned}$$

In other words, we would have that $\hat{x}^k = p_{R_n}f(\hat{x}^k)$. (In the final line above, we can return to $w \in \mathbb{R}^n$ by the definition of $\mathcal{L}_0$:

$$f(\hat{x}^k) \leq f(x^0) < f(x^0) + M_0 < f(w) < f(w) + R_n \frac{1}{2}|w - \hat{x}^k|^2$$

for $w \notin \mathcal{L}_0$.)

As usual in bundle methods, the convergence analysis considers two different asymptotic cases, depending on whether a finite or an infinite number of serious steps is done.

THEOREM 1 (asymptotic convergence of Algorithm 1). *Consider Algorithm* 1 *applied to a function f satisfying assumption* (1)$_{(x^0,M_0)}$, *with stopping parameter* $\mathrm{TOL_{stop}} = 0$ *and suppose there is no termination. Let $\bar\eta$ be the stabilized value for the convexification parameter sequence, as in Lemma* 3. *The following mutually exclusive situations hold:*

(a) *Either $\bar\eta > \rho^{id}$ and*

(a$_1$) *There is a last serious step $\hat{x}$, followed by infinitely many null steps. Then $x^{n+1} \to \hat{x}$, and $\hat{x}$ is a stationary point for f.*

(a$_2$) *There is an infinite number of serious steps. Then any accumulation point of the sequence $\{\hat{x}^k\}$ is a stationary point for f.*

(b) *Or $\bar\eta \leq \rho^{id}$.*

Proof. To see item (a$_1$), consider iterations n after the last serious step $\hat{x}$ was generated, so there are only null steps. We apply [12, Thm. 6] written x^0, V, R, and ρ therein replaced by $\hat{x}$, $\mathcal{L}_0$, $\bar{R}$, and ρ^{id}, respectively. We obtain that, as $n \to \infty$,

$$\text{the whole sequence } \{x^{n+1}\} \to p := p_{\bar{R}}f(\hat{x}) \text{ with } \varphi_n(x^{n+1}) \to f(p) + \frac{\bar\eta}{2}|p - \hat{x}|^2.$$

Thus,

$$\begin{array}{rcl}\delta_{n+1} & = & f(\hat{x}) + \bar{\eta}\tfrac{1}{2}|x^{n+1} - \hat{x}|^2 - \varphi_n(x^{n+1}) \\ & \to & f(\hat{x}) + \bar{\eta}\tfrac{1}{2}|p - \hat{x}|^2 - f(p) - \tfrac{\bar{\eta}}{2}|p - \hat{x}|^2 \\ & = & f(\hat{x}) - f(p).\end{array}$$

Since the serious step test in Step 3 of the algorithm is not satisfied, we have $f(x^{n+1}) > f(\hat{x}) - m\delta_{n+1}$. Taking the limit as $n \to \infty$ gives the relation $f(p) \geq f(\hat{x}) - m(f(\hat{x}) - f(p))$, so $f(\hat{x}) \leq f(p)$ because $m \in (0, 1)$. But $p = p_{\bar{R}}f(\hat{x})$ implies

$$f(p) + R\frac{1}{2}|p - \hat{x}|^2 \leq f(\hat{x}),$$

which shows that $\hat{x} = p$. That is, $\hat{x} = p_{\bar{R}}f(\hat{x})$, so $\hat{x}$ is a stationary point of f.

To see item (a$_2$), first notice that the sequence $\{\hat{x}^k\} \subset \mathcal{L}_0$, a compact set, so it has an accumulation point, say, for some infinite set K, $\hat{x}^k \to x^{\inf} \in \mathcal{L}_0$ as $K \ni k \to \infty$. Since $\hat{x}^{k+1} = x^{i_{k+1}}$, to alleviate notation we set $j_k = i_{k+1} - 1$ so that $\hat{x}^{k+1} = p_{\bar{\mu}}\varphi_{j_k}(\hat{x}^k)$. The telescopic sum of the descent test for the subsequence of serious steps

$$f(\hat{x}^{k+1}) \leq f(\hat{x}^k) - m\delta_{i_{k+1}}$$

implies that as $k \to \infty$, either $f(\hat{x}^k) \searrow -\infty$, or $\delta_{i_{k+1}} \to 0$. By Proposition 1(b), f is bounded below; therefore, $\delta_{i_{k+1}} \to 0$. From (17), this means that both $|\hat{x}^{k+1} - \hat{x}^k|^2$ and $e_{-j_k} + \bar{\eta}d^k_{-j_k}$ must converge to 0. Therefore, by (7), $\varphi_{j_k}(\hat{x}^{k+1}) - f(\hat{x}^k) \to 0$ as $k \to \infty$. Consider now $k \in K$. Since $|\hat{x}^{k+1} - \hat{x}^k|^2 \to 0$, both $\hat{x}^{k+1}$ and $\hat{x}^k$ converge to $x^{\inf}$ as $K \ni k \to \infty$ with $\varphi_{j_k}(\hat{x}^{k+1}) \to f(x^{\inf})$. But $\hat{x}^{k+1} = p_{\bar{\mu}}\varphi_{j_k}(\hat{x}^k)$ and $\bar{\eta} > \rho^{id}$ implies that for all $w \in \mathcal{L}_0$,

$$\varphi_{j_k}(\hat{x}^{k+1}) + \frac{\bar{\mu}}{2}|\hat{x}^{k+1} - \hat{x}^k|^2 \leq f(w) + \frac{\bar{R}}{2}|w - \hat{x}^k|^2,$$

by condition (19) (Lemma 3). Therefore, in the limit, we have that

$$f(x^{\inf}) \leq f(w) + \frac{\bar{R}}{2}|w - x^{\inf}|^2 \quad \text{for all } w \in \mathcal{L}_0.$$

As $x^{\inf} \in \mathcal{L}_0$, we also have that for any $w \notin \mathcal{L}_0$,

$$f(x^{\inf}) \leq f(x^0) + M_0 \leq f(w) \leq f(w) + \frac{\bar{R}}{2}|w - x^{\inf}|^2.$$

Hence,

$$f(x^{\inf}) \leq f(w) + \frac{\bar{R}}{2}|w - x^{\inf}|^2 \quad \text{for all } w \in \mathbb{R}^n.$$

In other words, $x^{\inf} = p_{\bar{R}}f(x^{\inf})$ with $\bar{R} > \rho^{id}$. Since f is lower-$\mathcal{C}^2$ at $x^{\inf}$ (as $x^{\inf} \in \mathcal{L}_0$) and $\bar{R} > \rho^{id}$, this implies that $0 \in \partial f(x^{\inf})$ [14, Prop. 2.1(g)]. □

Theorem 1 states convergence only in case (a), i.e., if the stabilized convexification parameter is greater than the ideal proximal threshold ρ^{id}. At this stage, case (b) cannot be ruled out. Indeed, it is conceivable to create an example where all generated iterates lie on a convex quadratic augmented function, yet the function itself is nonconvex. Details of such an example would be complicated to generate for the algorithm itself but would loosely look like the function drawn in Figure 1. Possible heuristics, modifying Algorithm 1 to address such difficulty, are discussed in the concluding section 7.

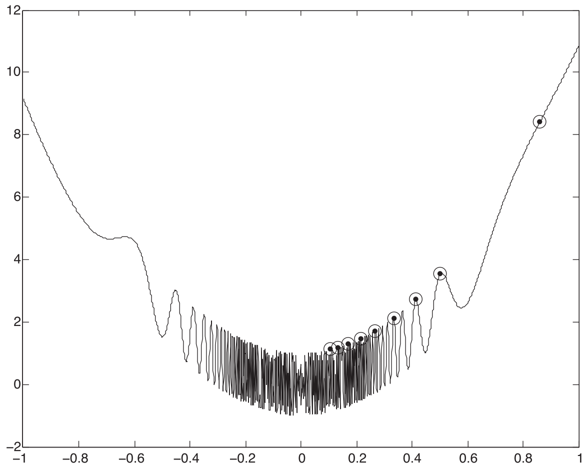

FIG. 1. *Example of how the stabilized convexification parameter might remain less than the ideal proximal threshold* ρ^{id}*.* $\left(f(x) = \sin(1/x^3) + 10x^2,\ x_i = (1/((\pi/2) + 2^i\pi i))^{1/3} \right)$

6. Numerical testing. In this section we explore some preliminary results on a numerical implementation of the redistributed bundle algorithm. The goal is to provide a proof-of-concept implementation, not a complete benchmarking of the algorithm. Nonetheless, in subsection 6.2, we provide a (very limited) comparison to two other existing bundle method solvers for nonconvex optimization.

6.1. Base implementation and tuning problems. Algorithm 1 was implemented in MATLAB v. 7.5.0.338 (R2007b). Two options were used to solve the QP in Step 1. First we solved Step 1 using QuadProg.m (Revision: 1.1.6.3), which is available in the Optimization Toolbox. Second we solved Step 1 using a MEX-interface for `qpdf2`, a Fortran code developed by K. C. Kiwiel for the method in [18].

In our first series of tests, we also seek to determine a default bundle maintenance and compare the two QP solvers `qpdf2` and QuadProg. Therefore, we attempted each test problem (discussed below) six times, using the two QP solvers discussed above and three different manners of maintaining the bundle. This provided six variants of Algorithm 1, given in Table 1 below.

TABLE 1
Algorithm 1 *variants.*

Variant	QP solver	Bundle I_{n+1}
1	`qpdf2`	$\{0, 1, 2, \ldots, n+1\}$
2	QuadProg	$\{0, 1, 2, \ldots, n+1\}$
3	`qpdf2`	$J_n^{act} \cup \{n+1, i_k\}$
4	QuadProg	$J_n^{act} \cup \{n+1, i_k\}$
5	`qpdf2`	$\{n+1, i_k, -n\}$
6	QuadProg	$\{n+1, i_k, -n\}$

We see that odd-numbered variants used `qpdf2` to solve the QP, while even-numbered variants used QuadProg.m.

As Algorithm 1 is a newly developed algorithm, there is no clear set of default parameters to use in the initialization step. Some parameters, such as the stopping tolerance, should always be defined by the user. For other parameters, such as the unacceptable increase parameter, it may be possible to develop reasonable default selections (keeping in mind that default selections may behave poorly on some problems). In particular, parameters R_0, M_0, m, and γ play a large role in the algorithm but have no clear directions for users to select specific values. The algorithm recalculates R based on function information, so setting $R_0 = 1$ or 10 seems as acceptable as any value. Values M_0, m, and γ are more likely to impact algorithmic performance. In our first series of tests, we seek to generate a reasonable set of default values for parameters M_0, m, and γ for the considered problems. To do this, a few options were attempted for each parameter: in particular, $M_0 = \{10, 10^3, 10^8\}$, $m = \{0.05, 0.1, 0.25\}$, and $\gamma = \{1.5, 2, 4\}$.

In order to obtain an indication of how these parameters (M_0, m, and γ) affect convergence rates, we considered a test set consisting of polynomial functions developed in [5]; see also [6]. For each $i = 1, 2, \ldots, N$, let the function h_i be defined via

$$\begin{aligned} h_i : \mathbb{R}^N &\mapsto \mathbb{R}, \\ x &\mapsto (ix_i^2 - 2x_i - K_1) + \textstyle\sum_{j=1}^N x_j, \end{aligned}$$

where K_1 is a fixed constant. Using the functions h_i, we define the five varieties of test functions via the following.

$$f_1(x) := \sum_{i=1}^N |h_i(x)|, \tag{21}$$

$$f_2(x) := \sum_{i=1}^N (h_i(x))^2, \tag{22}$$

$$f_3(x) := \max_{i \in \{1,2,\ldots,N\}} |h_i(x)|, \tag{23}$$

$$f_4(x) := \sum_{i=1}^N |h_i(x)| + \frac{1}{2}|x|^2, \tag{24}$$

$$f_5(x) := \sum_{i=1}^N |h_i(x)| + \frac{1}{2}|x|. \tag{25}$$

Figure 2 shows that these test functions are nonconvex in $\mathbb{R}^2$; they are all nonsmooth, except for f_2. These properties carry to higher dimensions as well.

In Lemma 2 we show that these test functions satisfy condition $(1)_{(0,M)}$ for any $M > 0$.

LEMMA 2 (properties of Ferrier polynomials). *Let f_1, f_2, f_3, f_4, and f_5 be defined as in* (21) *to* (25)*. Then each f_k ($k = 1$, 2, 3, 4, 5) is globally lower-$\mathcal{C}^2$, bounded below, and level coercive, and therefore the results of Example* 2 *apply. Moreover, if $K_1 = 0$, then*

$$0 = \min_x f_k \quad \textit{and} \quad \{0\} \in \operatorname{argmin}_x f_k \ \textit{for}\ k = 1, 2, 3, 4, 5.$$

Finally, $\{0\} = \operatorname{argmin} f_k(x)$ for $k = 4, 5$.

Proof. We begin by noting that the functions h_i are $\mathcal{C}^2$ and therefore lower-$\mathcal{C}^2$. Functions defined by sums, absolute values, maximums, and squares of lower-$\mathcal{C}^2$ functions are lower-$\mathcal{C}^2$ [34, Ex. 10.35, p. 452], and therefore each f_k is lower-$\mathcal{C}^2$. (The functions $|x|^2$ and $|x|$ are lower-$\mathcal{C}^2$ by the same principles.)

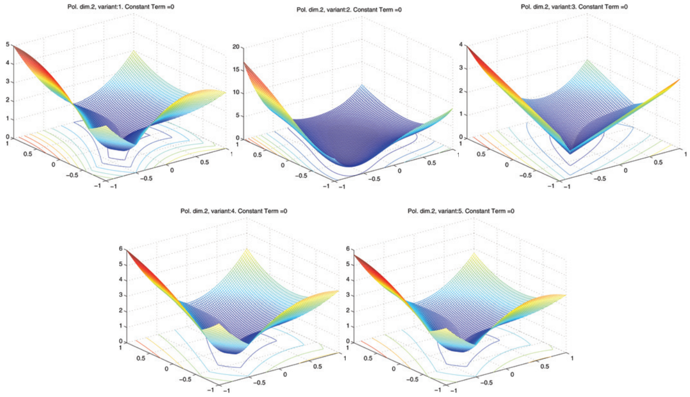

FIG. 2. *Ferrier polynomials (top-left), f_2 (top-middle), f_3 (top-right), f_4 (bottom-left), and f_5 (bottom-right) near* 0 *in* $\mathbb{R}^2$ *($K_1 = 0$).*

Each f_k is clearly bounded below by 0. The fact that each f_k is level coercive follows from having that each h_i grows quadratically in x_i and that each f_k is made via sums, absolute values, maximums, and squares for the h_i functions.

When $K_1 = 0$, we have $f_k(0) = 0$, so $0 = \min f_k(x)$ and $\{0\} \in \operatorname{argmin} f_k$. For $k = 4, 5$, the penalty term ($\frac{1}{2}|x|^2$ or $\frac{1}{2}|x|$) ensures $\{0\} = \operatorname{argmin} f_k(x)$. □

Remark 2. A quick estimate for the level of nonconvexity of f_1, f_2, f_3, and f_5 can be computed by noticing that nonconvexity arises from the $|h_i| = \max\{h_i, -h_i\}$. The Hessian of $-h_i$ can easily be computed to be the matrix with zeros in all entries except the i, i entry, which is $-2i$. As such, a prox-parameter $R_0 \geq 2N$ (where N is the dimension of the problem) will result in a well-defined proximal point.

Note that for $k \neq 4, 5$, we may have $\{0\} \supsetneq \operatorname{argmin} f_k$, as $f_1(1) = 0 = \min_x f_1$ when $N = 1$.

We considered 50 test problems with constant K_1 taken to be 0 and

$$\min_{x \in \mathbb{R}^N} f_k(x) \quad \text{for} \quad \begin{array}{l} N \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \\ k \in \{1, 2, 3, 4, 5\}. \end{array}$$

In examining the test set of Ferrier polynomials, we seek to generate a reasonable set of default values for parameters M_0, m, and γ. As mentioned, to do this we attempted each problem using the potential default parameters $M_0 = \{10, 10^3, 10^8\}$, $m = \{0.05, 0.1, 0.25\}$, and $\gamma = \{1.5, 2, 4\}$. For each test, the initial starting point was set to $\hat{x}^0 = [1, 1, \ldots, 1]$. To maintain a reasonable computation time, an upper limit of 300 function evaluations was implemented, and a stopping tolerance of $\text{TOL}_{\text{stop}} = 10^{-6}$ was applied. Summary results for all parameter selections appear in Tables 10 to 15 in Appendix A. The initial R value is set to $R_0 = 10$. The remaining parameters below represent those that provided the best results for this test set:

– the unacceptable increase parameter to $M_0 = 10$,
– the Armijo-like parameter to $m = 0.05$, and
– the convexification growth parameter to $\gamma = 2$.

Complete results for this parameter selection appear in Tables 4 to 9 in Appendix A. Letting x^* represent the point the algorithm returns as the most likely location of the local minimum, Tables 4 to 9 report the minimal function value found ($f^* = f(x^*)$), the value of δ_n at the final iteration (δ^*), and the number of oracle function evaluations used (fevals). Note that each time the oracle is called, both a function and gradient evaluation are made (i.e., gevals would equal fevals, so we do not report it).

Remark 3. Of course, an optimal parameter selection is highly dependent on the test problems examined, the starting point used, and the tolerance desired. There is no reason to believe that the parameter selection exploration above is anything more than the most preliminary analysis of reasonable default parameters.

6.2. Comparison with other methods. To provide a brief comparison of the redistributed bundle algorithm to other research, we also ran Algorithm 1 on 10 functions available in the literature and compared our results with those obtained by the variable metric nonconvex (VMNC) algorithm developed in [36] and the limited memory bundle method (LMBM) developed in [11].

Before proceeding, an important comment on nonsmooth optimization benchmarking is in order. Since the subdifferential of a nonsmooth function is not a continuous mapping, running the *same code* on different platforms can give *different results* because of slightly different responses from the oracle. For instance, for a *max*-type function, different machine precisions can lead to different subfunctions realizing the maximum and, hence, to (very) different subgradients. The situation is even more tricky when comparing different solvers: not only they should be run on the same platform, but they should all use the same f/g information for the oracle calculations. Our comparison was made on results obtained by the authors of each solver on different computers and with different coding for the oracle computations. For this reason, the comparison should be considered as just an indication of performance and not a real basis for evaluating the merits of the considered methods.

Each function's data and starting points are given in Table 2.

TABLE 2

Comparison test set information: function dimension (N), test starting point (x^0), and function minimum.

#	Name (f)	N	x^0	$\min f$
1	Crescent	2	$[-1.5, 2.0]$	0
2	Mifflin 2	2	$[-1.0, -1.0]$	-1
3	Colville 1	5	$[0, 0, 0, 0, 1]$	-32.348679
4	El-Attar	6	$[2, 2, 7, 0, -2, 1]$	0.5598131
5	Active Faces	2	$[1, 1]$	0
6	Active Faces	10	$[1, 1, \dots, 1]$	0
7	Active Faces	100	$[1, 1, \dots, 1]$	0
8	Brown	2	$[-1, 1]$	0
9	Brown	10	$[-1, 1, -1, 1, \dots, -1, 1]$	0
10	Brown	100	$[-1, 1, -1, 1, \dots, -1, 1]$	0

In Table 3 we present the results of the redistributed bundle (RedistProx) algorithm developed herein, the VMNC algorithm developed in [36], and the LMBM developed in [11]. RedistProx used the the parameter selection developed in subsection 6.1 ($\text{TOL}_{\text{stop}} = 10^{-6}$, $M_0 = 10$, $m = 0.05$, and $\gamma = 2$) using the QP solver `qpdf2`. The starting value R_0 was set to 10 for problems 1 to 4 and to 0.1 for problems 5 to 10. All tests terminated due to the stopping criterion in Step 2 of the algorithm. In particular, stoppage due to the maximum number of function evaluations was never invoked. The VMNC results are those reported in [36]. Only test problems 1 through 4 are reported in [36]. The LMBM results use parameters tuned and provided by LMBM's author, N. Karmitsa (née Haarala).

TABLE 3

Comparison results: RedistProx, VMNC [36]*, LMBM* [11]*, function evaluations used (fevals), and minimal objective value found (f^*).*

Function	RedistProx fevals	RedistProx f^*	VMNC fevals	VMNC f^*	LMBM fevals	LMBM f^*
1	39 ($R_0 = 10$)	0.466360E-07	15	0.949E-10	34	0.709715E-08
2	28 ($R_0 = 10$)	-0.9999993	35	-0.9999998	30	-0.9999981
3	41 ($R_0 = 10$)	-32.348673	47	-32.348675	90	-32.348524
4	91 ($R_0 = 10$)	0.5598162	76	0.5598184	232	0.5598160
5	10 ($R_0 = 0.1$)	6.612847E-009	NA	NA	14	1.880629E-011
6	14 ($R_0 = 0.1$)	1.086686E-012	NA	NA	24	8.881784E-016
7	20 ($R_0 = 0.1$)	7.785328E-012	NA	NA	64	8.171241E-014
8	11 ($R_0 = 0.1$)	2.547046E-011	NA	NA	12	1.309847E-016
9	20 ($R_0 = 0.1$)	1.302742E-009	NA	NA	67	3.950421E-010
10	31 ($R_0 = 0.1$)	6.785571E-007	NA	NA	93	4.219298E-009

We see that an optimal solution to accuracy 10^{-6} was obtained by all algorithms on all problems, except problem 4, El-Attar, where an optimal solution with accuracy of 10^{-5} was obtained. The RedistProx algorithm compares quite well to VMNC, doing better on some problems and worse on others. The RedistProx algorithm uses less function evaluations than the LMBM on most problems. However, it should be noted that the LMBM was designed for high-dimension problems, which are not considered in our comparisons.

7. Conclusion. We have presented a novel proximal bundle algorithm that is designed to work on nonconvex functions. Algorithmic convergence is studied for the class of function defined in (1), which includes all lower-$\mathcal{C}^2$ functions with at least one bounded level set.

The algorithm provides somes ingredients having a different flavor over previous bundle methods for nonconvex functions. First and foremost, the algorithm is designed from both primal and dual perspectives, whose theoretical foundation should help in improving future algorithms of the proximal type. Another advantage, not explored in this paper, is that the algorithm provides feedback on the level of convexification required to make the proximal point of the model function unique. It is likely that this value also provides feedback on the level of convexification required to make the objective function locally convex. This is open to future research.

Analysis of the convergence of the algorithm proceeded by first showing that the convexification parameter eventually stabilized. Once stabilized, convergence was proven under the assumption that the stabilized convexification parameter is greater than the ideal proximal threshold ρ^{id}. The example in Figure 1 shows a situation where such an assumption may not be true. Generating a precise example like that shown in Figure 1 would likely be more complicated than it appears, and it seems unlikely to arise in actual practice. It would be interesting to know if a simple example could be generated where situation (b) of Theorem 1 arises. A simple adaptation of the algorithm which would aid in avoiding situation (b) of Theorem 1 would be to add a small random element to the solution of the QP subproblem (Algorithm 1, Step 1). In particular, if $x^{n+1} = p_{\mu_n}\varphi_{0,\eta_0}(\hat{x}^k)$, then the next iterate point could be selected as $x^{n+1} + \epsilon_{n+1}$, where ϵ_{n+1} is a random vector of diminishing norm. Although convergence results will not trivially carry over for such an algorithm, by the classical robustness of the proximal point method (see [14]) and of lower-$\mathcal{C}^2$ functions (which are locally Lipschitz), convergence results should remain stable. Such an algorithm should eventually either find a point which forces the convexification parameter to be greater than ρ^{id} or converge to a local minimum where the convexification parameter is larger than the local ideal proximal threshold. Further research will explore this idea.

Preliminary numerical results are very promising. The RedistProx algorithm compares well to two known nonconvex bundle methods, VMNC and LMBM, on a collection of 10 nonconvex problems. By examining Ferrier polynomials of dimensions 1 through 10, we developed a reasonable set of default parameters for the algorithm. These polynomials provide an easy-to-use collection of nonconvex functions with which to work. In addition we considered three potential bundle cleaning methods and two potential QP solvers. The results of the best-found parameter selection, bundle cleaning strategy, and QP solver appear in Table 8. This combination solved 48 of the 50 tuning problems to an objective function value of less than 0.05. Detailed work on determining good initialization parameters will likely further increase algorithmic performance.

Appendix A. Tables with results.

TABLE 4

Results of variant 1: $I_{n+1} = \{0, 1, 2, \ldots, n+1\}$, *qpsolver* = `qpdf2`.

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.000000	0.000000	49
2	1	0.000000	0.000000	2	2	6	0.000000	0.000133	36
3	1	0.000000	0.000000	2	3	6	0.093031	0.218325	26
4	1	0.000000	0.000534	28	4	6	0.022327	0.129176	55
5	1	0.000000	0.000007	13	5	6	0.113834	0.217826	37
1	2	0.086533	0.196793	12	1	7	0.167794	0.175417	36
2	2	0.000001	0.001090	22	2	7	0.000000	0.000166	48
3	2	0.000000	0.000043	15	3	7	0.018131	0.299913	30
4	2	0.036623	0.080019	10	4	7	0.239281	0.257493	34
5	2	0.000000	0.000000	16	5	7	0.099638	0.161789	84
1	3	0.000000	0.000052	19	1	8	0.172942	0.000594	301
2	3	0.000000	0.000371	31	2	8	0.030530	0.000110	124
3	3	0.000320	0.007858	16	3	8	0.067895	0.073139	34
4	3	0.056012	0.249029	14	4	8	0.809388	0.206697	38
5	3	0.000000	0.000003	16	5	8	1.084600	0.086655	40
1	4	0.074170	0.149659	17	1	9	0.000001	0.000035	67
2	4	0.000000	0.000146	33	2	9	0.000000	0.000000	37
3	4	0.007471	0.417435	16	3	9	0.000591	0.032509	48
4	4	0.025722	0.123606	19	4	9	0.038801	0.109996	301
5	4	0.019105	0.128998	23	5	9	0.769331	0.265637	40
1	5	0.213263	0.247224	23	1	10	0.043488	0.457050	301
2	5	0.000000	0.000122	56	2	10	0.000000	0.000088	36
3	5	0.013662	0.680244	21	3	10	0.021357	0.032731	43
4	5	0.051166	0.304311	26	4	10	0.141920	0.227919	94
5	5	0.351331	0.374701	32	5	10	0.164544	0.212437	69

Table 5

Results of variant 2: $I_{n+1} = \{0, 1, 2, \dots, n+1\}$, qpsolver = QuadProg.m.

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.000000	0.000000	49
2	1	0.000000	0.000000	2	2	6	58.250090	1.132209	301
3	1	0.000000	0.000000	2	3	6	0.093031	0.218325	26
4	1	0.500000	0.000010	5	4	6	0.000000	0.000020	60
5	1	0.000000	0.000007	13	5	6	0.113835	0.217732	35
1	2	0.086533	0.196793	12	1	7	0.146978	0.186392	34
2	2	0.000005	0.002908	16	2	7	0.000000	0.000055	162
3	2	0.000000	0.000655	14	3	7	0.000051	0.018630	33
4	2	0.036623	0.080019	10	4	7	0.239281	0.257493	34
5	2	0.000000	0.000000	16	5	7	0.120009	0.182087	41
1	3	0.001243	0.029610	13	1	8	0.335781	0.280330	45
2	3	0.000000	0.000158	26	2	8	0.000000	0.000002	168
3	3	0.000320	0.007858	16	3	8	0.067895	0.073139	34
4	3	0.052922	0.252121	12	4	8	0.069823	0.154684	56
5	3	0.000000	0.000002	17	5	8	1.084600	0.086662	41
1	4	0.000000	0.000003	26	1	9	0.000565	0.027483	74
2	4	11.986650	0.942896	301	2	9	146.779005	1.663763	301
3	4	0.001048	0.432945	17	3	9	0.070714	0.263645	42
4	4	0.025722	0.123599	18	4	9	0.000000	0.000014	150
5	4	0.019105	0.128998	23	5	9	0.782526	0.238298	42
1	5	0.213263	0.247224	23	1	10	0.913725	0.107716	301
2	5	26.972796	1.077798	301	2	10	303.451708	1.529603	301
3	5	0.013663	0.680228	20	3	10	0.028954	0.317898	46
4	5	0.051166	0.304311	26	4	10	0.217352	0.171962	61
5	5	0.351332	0.374674	31	5	10	0.036327	0.087404	66

Table 6

Results of variant 3: $I_{n+1} = J_n^{act} \cup \{n+1, i_k\}$, qpsolver = `qpdf2`.

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.000000	0.000007	41
2	1	0.000000	0.000000	2	2	6	0.000000	0.000174	35
3	1	0.000000	0.000000	2	3	6	0.093031	0.218345	25
4	1	0.000000	0.000534	28	4	6	0.022756	0.130716	41
5	1	0.000000	0.000007	13	5	6	0.113377	0.225677	39
1	2	0.086533	0.196793	12	1	7	0.364917	0.345851	36
2	2	0.000001	0.001090	22	2	7	0.000000	0.000143	51
3	2	0.000000	0.000270	15	3	7	0.018131	0.299913	30
4	2	0.036623	0.080019	10	4	7	0.102991	0.000004	55
5	2	0.000000	0.000001	17	5	7	0.088420	0.143231	50
1	3	0.000726	0.013312	13	1	8	0.379768	0.336665	39
2	3	0.000000	0.000289	31	2	8	0.074621	0.147004	59
3	3	0.000467	0.032766	14	3	8	0.067895	0.073184	35
4	3	0.056012	0.249029	14	4	8	0.797753	0.234319	32
5	3	0.000000	0.000003	16	5	8	1.077575	0.142200	38
1	4	0.074170	0.149659	17	1	9	0.082569	0.247277	301
2	4	0.000000	0.000146	33	2	9	0.000000	0.000115	45
3	4	0.001048	0.432945	17	3	9	0.000591	0.032312	301
4	4	0.025722	0.123606	19	4	9	0.000584	0.016981	61
5	4	0.019105	0.128998	23	5	9	0.807309	0.204687	53
1	5	0.213263	0.247224	23	1	10	0.106221	0.160658	52
2	5	0.000000	0.000329	85	2	10	0.000000	0.000088	36
3	5	0.013662	0.680244	21	3	10	0.016641	0.785673	39
4	5	0.051231	0.294591	28	4	10	0.002794	0.046028	67
5	5	0.352803	0.356526	26	5	10	0.167013	0.197659	70

Table 7

Results of variant 4*:* $I_{n+1} = J_n^{act} \cup \{n+1, i_k\}$*, qpsolver = QuadProg.m.*

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.000436	0.001865	33
2	1	0.000000	0.000000	2	2	6	0.000000	0.000262	188
3	1	0.000000	0.000000	2	3	6	0.093031	0.218345	25
4	1	0.500000	0.000010	5	4	6	0.000141	0.001911	48
5	1	0.000000	0.000007	13	5	6	0.113377	0.225676	301
1	2	0.086533	0.196793	12	1	7	0.348785	0.333532	36
2	2	0.000005	0.002908	16	2	7	0.000017	0.000099	204
3	2	0.000000	0.000322	14	3	7	0.000017	0.002823	38
4	2	0.036623	0.080019	10	4	7	0.102991	0.000184	301
5	2	0.000000	0.000001	17	5	7	0.021601	0.055467	301
1	3	0.001243	0.029610	13	1	8	0.379767	0.337170	301
2	3	0.000000	0.000158	26	2	8	0.000024	0.000179	301
3	3	0.000467	0.032766	14	3	8	0.067895	0.073184	35
4	3	0.052922	0.252121	12	4	8	0.005559	0.134004	301
5	3	0.000000	0.000002	17	5	8	1.077575	0.142200	38
1	4	0.000019	0.000399	22	1	9	0.082567	0.247492	301
2	4	0.000000	0.000232	68	2	9	0.000917	0.002066	301
3	4	0.001048	0.432945	17	3	9	0.070714	0.263612	40
4	4	0.025722	0.123599	18	4	9	0.010826	0.011645	123
5	4	0.019105	0.128998	23	5	9	0.807309	0.204719	301
1	5	0.213263	0.247224	23	1	10	0.000010	0.000118	85
2	5	0.000000	0.000174	135	2	10	0.003601	0.001106	301
3	5	0.013663	0.680228	20	3	10	0.028953	0.317898	43
4	5	0.051231	0.294591	28	4	10	0.165442	0.204766	46
5	5	0.352803	0.356517	29	5	10	0.036327	0.087441	301

Table 8

Results of variant 5*:* $I_{n+1} = \{n+1, i_k, -n\}$*, qpsolver =* `qpdf2`*.*

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.122881	0.007125	301
2	1	0.000000	0.000000	2	2	6	0.000000	0.000143	45
3	1	0.000000	0.000000	2	3	6	0.001095	0.041370	97
4	1	0.000000	0.000284	29	4	6	0.000003	0.044297	301
5	1	0.000000	0.000007	13	5	6	0.015945	0.007109	301
1	2	0.000031	0.001141	301	1	7	0.004317	0.009077	301
2	2	0.000001	0.001090	22	2	7	0.000000	0.000165	69
3	2	0.000001	0.000094	27	3	7	0.001862	0.000892	301
4	2	0.000188	0.002720	301	4	7	0.008193	0.043110	100
5	2	0.009355	0.035699	29	5	7	0.037477	0.066359	121
1	3	0.000813	0.011525	30	1	8	0.000008	0.000132	301
2	3	0.000000	0.000075	74	2	8	0.030530	0.000110	190
3	3	0.000001	0.000003	83	3	8	0.000030	0.000010	301
4	3	0.000001	0.000084	67	4	8	0.000021	0.000107	301
5	3	0.009583	0.109821	27	5	8	0.000534	0.003287	301
1	4	0.004490	0.004226	301	1	9	0.000671	0.000762	301
2	4	0.000000	0.000182	18	2	9	0.000000	0.000223	78
3	4	0.000002	0.000991	34	3	9	0.034807	0.033207	301
4	4	0.000005	0.000682	301	4	9	0.001084	0.007102	301
5	4	0.010121	0.011652	301	5	9	0.002150	0.012331	117
1	5	0.000001	0.000003	152	1	10	0.001967	0.000479	301
2	5	0.000000	0.000175	49	2	10	0.000000	0.000115	38
3	5	0.000001	0.000001	261	3	10	0.069996	0.037043	301
4	5	0.000036	0.003798	186	4	10	0.003564	0.023635	160
5	5	0.002217	0.025979	91	5	10	0.013286	0.032663	155

TABLE 9
Results of variant 6: $I_{n+1} = \{n+1, i_k, -n\}$, qpsolver = QuadProg.m.

k	n	f^*	δ^*	fevals	k	n	f^*	δ^*	fevals
1	1	0.000000	0.000000	2	1	6	0.007279	0.028079	301
2	1	0.000000	0.000000	2	2	6	58.250093	1.132209	301
3	1	0.000000	0.000000	2	3	6	0.001095	0.041370	97
4	1	0.500000	0.000010	5	4	6	0.000002	0.000001	289
5	1	0.000943	0.023876	10	5	6	0.053403	0.013682	301
1	2	0.000031	0.001141	301	1	7	0.000941	0.014519	301
2	2	0.000010	0.000000	18	2	7	32.247554	0.625927	301
3	2	0.000001	0.000420	25	3	7	0.002286	0.001549	301
4	2	0.000188	0.002720	301	4	7	0.008876	0.037044	103
5	2	0.000039	0.001036	301	5	7	0.018334	0.130957	67
1	3	0.002374	0.045005	28	1	8	0.000065	0.001373	187
2	3	0.000001	0.000339	33	2	8	7.055962	0.412997	301
3	3	0.000000	0.000001	86	3	8	0.000024	0.000015	301
4	3	0.000006	0.002360	90	4	8	0.000043	0.000240	301
5	3	0.009583	0.109821	27	5	8	0.001695	0.016130	301
1	4	0.000002	0.000760	268	1	9	0.000376	0.000264	301
2	4	0.000001	0.000445	108	2	9	146.779007	1.663763	301
3	4	0.000002	0.000991	34	3	9	0.000010	0.006468	193
4	4	0.000005	0.000682	301	4	9	0.281491	0.020101	301
5	4	0.010516	0.012299	301	5	9	0.005104	0.001976	301
1	5	0.000001	0.000003	149	1	10	0.040198	0.113631	301
2	5	26.972797	1.077798	301	2	10	303.451704	1.529603	301
3	5	0.000001	0.000001	261	3	10	0.002658	0.073758	154
4	5	7.948708	1.179115	3	4	10	0.426285	0.056585	301
5	5	0.002217	0.025979	91	5	10	0.215707	0.203468	186

TABLE 10
Summary of parameter testing of variant 1: $I_{n+1} = \{0, 1, 2, \ldots, n+1\}$, qpsolver = `qpdf2`.

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	28	18	15	43.3 (48.0)
10	0.05	2	30	21	17	42.7 (56.3)
10	0.05	4	30	22	20	31.3 (18.8)
10	0.1	1.5	32	20	15	49.4 (68.5)
10	0.1	2	33	25	21	33.1 (23.6)
10	0.1	4	30	23	20	29.4 (17.0)
10	0.25	1.5	28	17	16	30.6 (17.1)
10	0.25	2	26	17	16	31.0 (19.5)
10	0.25	4	29	19	18	29.6 (18.8)
10^3	0.05	1.5	31	19	16	43.9 (47.6)
10^3	0.05	2	31	18	15	44.9 (51.9)
10^3	0.05	4	31	19	17	34.0 (21.2)
10^3	0.1	1.5	34	22	17	49.6 (60.9)
10^3	0.1	2	31	22	16	41.9 (45.1)
10^3	0.1	4	31	21	17	42.5 (47.2)
10^3	0.25	1.5	26	16	15	38.6 (43.0)
10^3	0.25	2	26	16	15	32.9 (19.9)
10^3	0.25	4	28	16	14	33.0 (20.4)
10^8	0.05	1.5	31	19	16	54.9 (68.1)
10^8	0.05	2	32	19	16	55.7 (68.3)
10^8	0.05	4	31	18	16	49.1 (52.8)
10^8	0.1	1.5	35	24	19	55.1 (65.3)
10^8	0.1	2	33	24	19	49.6 (53.6)
10^8	0.1	4	34	24	20	50.2 (50.4)
10^8	0.25	1.5	26	16	15	53.7 (67.5)
10^8	0.25	2	26	16	15	47.1 (51.8)
10^8	0.25	4	28	15	14	48.3 (55.4)

Table 11

Summary of parameter testing of variant 2: $I_{n+1} = \{0, 1, 2, \ldots, n+1\}$, qpsolver = QuadProg.m.

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	29	18	13	98.7 (117.6)
10	0.05	2	33	19	13	95.2 (115.8)
10	0.05	4	27	18	13	96.6 (117.7)
10	0.1	1.5	29	16	11	88.0 (107.7)
10	0.1	2	28	18	11	103.9 (121.7)
10	0.1	4	29	15	11	86.1 (107.1)
10	0.25	1.5	31	11	6	116.8 (119.3)
10	0.25	2	30	11	7	137.4 (128.1)
10	0.25	4	31	13	7	121.6 (118.9)
10^3	0.05	1.5	30	19	15	93.1 (112.4)
10^3	0.05	2	34	20	15	103.4 (119.7)
10^3	0.05	4	32	18	14	91.7 (112.4)
10^3	0.1	1.5	27	17	14	96.7 (114.9)
10^3	0.1	2	28	18	13	93.3 (111.5)
10^3	0.1	4	27	17	13	96.4 (114.3)
10^3	0.25	1.5	31	14	8	106.8 (112.1)
10^3	0.25	2	31	15	6	116.4 (116.4)
10^3	0.25	4	30	16	8	98.9 (105.9)
10^8	0.05	1.5	26	15	11	86.5 (110.8)
10^8	0.05	2	30	16	11	96.2 (118.3)
10^8	0.05	4	28	15	11	81.7 (106.5)
10^8	0.1	1.5	23	14	11	84.7 (111.0)
10^8	0.1	2	24	14	10	80.1 (106.6)
10^8	0.1	4	23	14	11	80.6 (106.4)
10^8	0.25	1.5	27	11	8	94.5 (107.2)
10^8	0.25	2	27	12	8	104.3 (112.7)
10^8	0.25	4	27	14	10	85.1 (96.4)

TABLE 12
Summary of parameter testing of variant 3: $I_{n+1} = J_n^{act} \cup \{n+1, i_k\}$, *qpsolver* = `qpdf2`.

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	29	17	15	60.1 (79.5)
10	0.05	2	33	22	20	45.9 (58.4)
10	0.05	4	29	21	20	39.0 (45.4)
10	0.1	1.5	31	19	17	41.6 (49.2)
10	0.1	2	34	26	23	40.8 (45.9)
10	0.1	4	29	21	20	30.2 (19.1)
10	0.25	1.5	32	17	16	34.3 (34.4)
10	0.25	2	28	16	15	32.5 (23.1)
10	0.25	4	31	19	18	32.5 (24.7)
10^3	0.05	1.5	31	19	17	43.5 (38.4)
10^3	0.05	2	31	18	15	45.5 (46.6)
10^3	0.05	4	30	20	18	42.4 (45.1)
10^3	0.1	1.5	34	19	17	51.2 (65.0)
10^3	0.1	2	32	19	15	48.6 (59.6)
10^3	0.1	4	31	19	18	45.4 (53.8)
10^3	0.25	1.5	29	17	16	34.5 (24.8)
10^3	0.25	2	29	17	16	34.9 (25.1)
10^3	0.25	4	30	18	16	35.3 (26.9)
10^8	0.05	1.5	31	17	16	54.5 (60.0)
10^8	0.05	2	32	18	16	53.4 (55.9)
10^8	0.05	4	30	18	17	56.7 (63.2)
10^8	0.1	1.5	35	21	19	55.8 (67.1)
10^8	0.1	2	34	21	18	55.4 (65.0)
10^8	0.1	4	33	21	19	54.5 (62.5)
10^8	0.25	1.5	29	15	14	49.5 (56.3)
10^8	0.25	2	29	15	14	49.5 (55.5)
10^8	0.25	4	30	16	15	48.9 (50.9)

TABLE 13

Summary of parameter testing of variant 4: $I_{n+1} = J_n^{act} \cup \{n+1, i_k\}$*, qpsolver = QuadProg.m.*

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	21	13	10	75.3 (103.1)
10	0.05	2	24	16	10	74.5 (102.6)
10	0.05	4	21	13	9	76.7 (102.6)
10	0.1	1.5	24	11	9	81.2 (106.2)
10	0.1	2	20	9	7	79.0 (106.7)
10	0.1	4	21	8	7	80.6 (106.5)
10	0.25	1.5	15	5	4	134.0 (133.1)
10	0.25	2	14	3	3	168.3 (139.9)
10	0.25	4	12	3	3	168.3 (139.9)
10^3	0.05	1.5	24	13	10	64.7 (90.4)
10^3	0.05	2	26	14	10	59.7 (84.1)
10^3	0.05	4	25	13	10	67.7 (91.9)
10^3	0.1	1.5	23	10	8	76.0 (102.0)
10^3	0.1	2	21	10	8	76.9 (102.0)
10^3	0.1	4	22	11	8	77.5 (102.3)
10^3	0.25	1.5	15	6	4	128.0 (128.1)
10^3	0.25	2	16	5	4	128.8 (127.9)
10^3	0.25	4	16	7	6	128.1 (128.0)
10^8	0.05	1.5	30	18	14	52.0 (65.0)
10^8	0.05	2	31	18	14	50.7 (60.1)
10^8	0.05	4	31	18	15	51.0 (58.4)
10^8	0.1	1.5	30	16	13	60.7 (79.0)
10^8	0.1	2	27	15	12	61.9 (80.0)
10^8	0.1	4	28	16	12	62.5 (80.5)
10^8	0.25	1.5	19	6	4	128.0 (128.1)
10^8	0.25	2	19	5	4	128.8 (127.9)
10^8	0.25	4	19	7	6	128.1 (128.0)

TABLE 14
Summary of parameter testing of variant 5: $I_{n+1} = \{n+1, i_k, -n\}$, *qpsolver* = `qpdf2`.

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	47	35	19	182.3 (124.6)
10	0.05	2	48	30	18	161.8 (121.4)
10	0.05	4	47	32	18	173.5 (128.1)
10	0.1	1.5	48	31	18	175.3 (124.9)
10	0.1	2	48	32	16	165.4 (126.3)
10	0.1	4	46	28	17	162.4 (127.5)
10	0.25	1.5	41	27	18	178.9 (130.8)
10	0.25	2	46	34	17	170.4 (127.4)
10	0.25	4	45	32	18	169.6 (126.6)
10^3	0.05	1.5	41	25	18	188.7 (121.7)
10^3	0.05	2	37	25	18	179.2 (124.5)
10^3	0.05	4	37	27	17	184.8 (127.3)
10^3	0.1	1.5	42	26	18	183.8 (123.2)
10^3	0.1	2	37	22	15	186.7 (126.7)
10^3	0.1	4	37	22	15	182.8 (129.3)
10^3	0.25	1.5	41	25	18	185.5 (130.4)
10^3	0.25	2	40	25	16	188.1 (130.8)
10^3	0.25	4	39	26	17	186.4 (129.6)
10^8	0.05	1.5	38	21	15	201.3 (122.6)
10^8	0.05	2	34	21	14	197.8 (124.9)
10^8	0.05	4	35	25	15	204.3 (122.4)
10^8	0.1	1.5	39	23	14	201.7 (124.2)
10^8	0.1	2	35	21	13	205.4 (125.0)
10^8	0.1	4	35	22	14	202.9 (124.0)
10^8	0.25	1.5	38	22	15	208.8 (123.7)
10^8	0.25	2	37	22	13	212.4 (122.5)
10^8	0.25	4	37	24	14	210.8 (121.3)

Table 15
Summary of parameter testing of variant 6*:* $I_{n+1} = \{n+1, i_k, -n\}$, *qpsolver = QuadProg.m.*

M_0	m	γ	# solved to $f^* < 0.05$	# solved to $f^* < 10^{-3}$	# solved to $f^* < 10^{-6}$	average (std) feval
10	0.05	1.5	39	21	9	204.2 (122.6)
10	0.05	2	37	23	8	197.3 (123.0)
10	0.05	4	36	22	7	202.3 (122.4)
10	0.1	1.5	32	17	8	196.9 (122.6)
10	0.1	2	33	19	10	208.9 (122.2)
10	0.1	4	33	16	7	204.2 (119.4)
10	0.25	1.5	25	10	5	245.8 (107.6)
10	0.25	2	24	10	6	239.7 (109.8)
10	0.25	4	20	11	5	236.4 (108.0)
10^3	0.05	1.5	35	18	7	208.8 (123.3)
10^3	0.05	2	33	18	7	203.1 (125.4)
10^3	0.05	4	31	20	7	207.2 (121.9)
10^3	0.1	1.5	32	16	8	206.1 (126.4)
10^3	0.1	2	29	16	8	213.0 (126.1)
10^3	0.1	4	29	16	7	224.4 (121.5)
10^3	0.25	1.5	23	10	4	244.0 (110.9)
10^3	0.25	2	26	12	5	238.1 (112.7)
10^3	0.25	4	23	12	5	235.0 (110.6)
10^8	0.05	1.5	38	21	9	202.9 (122.5)
10^8	0.05	2	36	21	9	197.5 (124.0)
10^8	0.05	4	35	24	10	190.6 (123.9)
10^8	0.1	1.5	36	20	12	194.5 (126.4)
10^8	0.1	2	33	20	12	201.5 (126.8)
10^8	0.1	4	32	20	11	206.9 (126.1)
10^8	0.25	1.5	22	10	4	244.0 (110.9)
10^8	0.25	2	24	10	5	238.1 (112.7)
10^8	0.25	4	22	11	5	229.0 (114.9)

Acknowledgments. We would like to thank the IRMACS Centre at Simon Fraser University, and in particular Brian Corrie, for technical assistance during the numerical testing presented within this work. We are also grateful to Napsu Karmitsa for providing the LMBM results in Table 3, as well as the two reviewers for valuable comments and suggestions.

REFERENCES

[1] A. M. Bagirov, B. Karasözen, and M. Sezer, *Discrete gradient method: derivative-free method for nonsmooth optimization*, J. Optim. Theory Appl., 137 (2008), pp. 317–334.

[2] A. M. Bagirov and J. Yearwood, *A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems*, European J. Oper. Res., 170 (2006), pp. 578–596.

[3] J. V. Burke, A. S. Lewis, and M. L. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., 15 (2005), pp. 751–779.

[4] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Program., 62 (1993), pp. 261–275.

[5] C. Ferrier, *Bornes Duales de Problémes d'Optimisation Polynomiaux*, Ph.D. thesis, Laboratoire Approximation et Optimisation, Université Paul Sabatier, Toulouse, France, 1997.

[6] C. Ferrier, *Computation of the distance to semi-algebraic sets*, ESAIM Control Optim. Calc. Var., 5 (2000), pp. 139–156.

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

[8] A. Fuduli, M. Gaudioso, and G. Giallombardo, *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control*, SIAM J. Optim., 14 (2004), pp. 743–756.

[9] A. Fuduli, M. Gaudioso, and G. Giallombardo, *A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization*, Optim. Methods Softw., 19 (2004), pp. 89–102.

[10] A. M. Gupal, *A method of for the minimization of almost differentiable functions*, Kibernetika (Kiev), 1 (1977), pp. 114–116.

[11] N. Haarala, K. Miettinen, and M. M. Mäkelä, *Globally convergent limited memory bundle method for large-scale nonsmooth optimization*, Math. Program., 109 (2007), pp. 181–205.

[12] W. Hare and C. Sagastizábal, *Computing proximal points of nonconvex functions*, Math. Program., 116 (2009), pp. 221–258.

[13] W. L. Hare and A. S. Lewis, *Identifying active constraints via partial smoothness and prox-regularity*, J. Convex Anal., 11 (2004), pp. 251–266.

[14] W. L. Hare and R. A. Poliquin, *Prox-regularity and stability of the proximal mapping*, J. Convex Anal., 14 (2007), pp. 589–606.

[15] J.-B. Hiriart-Urruty, *Generalized differentiability, duality and optimization for problems dealing with differences of convex functions*, in Convexity and Duality in Optimization Lecture Notes in Econom. and Math. Systems 256, Springer, Berlin, 1985, pp. 37–70.

[16] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms.* II, in Grundlehren der Mathematischen Wissenschaften 306 [Fundamental Principles of Mathematical Sciences 306], Springer, Berlin, 1993.

[17] K. C. Kiwiel, *A linearization algorithm for nonsmooth minimization*, Math. Oper. Res., 10 (1985), pp. 185–194.

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

[19] K. C. Kiwiel, *Restricted step and Levenberg–Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization*, SIAM J. Optim., 6 (1996), pp. 227–249.

[20] K. C. Kiwiel, *A method of centers with approximate subgradient linearizations for nonsmooth convex optimization*, SIAM J. Optim., 18 (2008), pp. 1467–1489.

[21] C. Lemaréchal, *An extension of Davidon methods to nondifferentiable problems*, Math. Programming Stud., 3 (1975), pp. 95–109.

[22] C. Lemaréchal, *Bundle methods in nonsmooth optimization*, in Nonsmooth Optimization, Proceedings of the International Institute for Applied Systems Analysis, 3, Laxenburg, Austria, Pergamon, Oxford, 1978, pp. 79–102.

[23] C. Lemaréchal, *Lagrangian relaxation*, in Computational Combinatorial Optimization, Lecture Notes in Comput. Sci. 2241, Springer, Berlin, 2001, pp. 112–156.

[24] C. Lemaréchal, J.-J. Strodiot, and A. Bihain, *On a bundle algorithm for nonsmooth optimization*, in Proceedings of the Fourth Nonlinear Programming Symposium, Madison, WI, Academic Press, New York, 1981, pp. 245–282.

[25] L. Lukšan and J. Vlček, *A bundle-Newton method for nonsmooth unconstrained minimization*, Math. Program., 83 (1998), pp. 373–391.

[26] M. M. Mäkelä and P. Neittaanmäki, *Nonsmooth Optimization*, World Scientific Publishing Co., River Edge, NJ, 1992.

[27] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.

[28] R. Mifflin, *Convergence of a modification of Lemaréchal's algorithm for nonsmooth optimization*, in Progress in Nondifferentiable Optimization, IIASA Collaborative Proceedings Series, 8, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982, pp. 85–95.

[29] R. Mifflin, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Math. Programming Stud., 17 (1982), pp. 77–90.

[30] R. Mifflin and C. Sagastizábal, *Primal-dual gradient structured functions: Second-order results; links to epi-derivatives and partly smooth functions*, SIAM J. Optim., 13 (2003), pp. 1174–1194.

[31] R. Mifflin and C. Sagastizábal, *$\mathcal{VU}$-smoothness and proximal point results for some nonconvex functions*, Optim. Methods Softw., 19 (2004), pp. 463–478.

[32] D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim., 4 (2008), pp. 569–602.

[33] R. A. Poliquin and R. T. Rockafellar, *Prox-regular functions in variational analysis*, Trans. Amer. Math. Soc., 348 (1996), pp. 1805–1838.

[34] R. T. Rockafellar and J. J.-B. Wets, *Variational Analysis*, in Grundlehren der Mathematischen Wissenschaften, 317 [Fundamental Principles of Mathematical Sciences, 317], Springer, Berlin, 1998.

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

[36] J. Vlček and L. Lukšan, *Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization*, J. Optim. Theory Appl., 111 (2001), pp. 407–430.