

# A proximal-projection bundle method for convex nonsmooth optimization with on-demand accuracy oracles

Xiaoxia Dong, Chunming Tang\*, Haiyan Zheng

*College of Mathematics and Information Science, Guangxi University, Nanning 530004, P. R. China.*

**Abstract** For some practical problems, the exact computation of the function and (sub)gradient values may be difficult. In this paper, a proximal-projection bundle method for minimizing convex nonsmooth optimization problems with on-demand accuracy oracles is proposed. Our method essentially generalizes the work of Kiwiel (*SIAM J Optim*, 17: 1015-1034, 2006) from exact and inexact oracles to various oracles, including exact, inexact, partially inexact, asymptotically exact and partially asymptotically exact oracles. At each iteration, a proximal subproblem is solved to generate a linear model of the objective function, and then a projection subproblem is solved to obtain a trial point. Finally, global convergence of the algorithm is established under different types of inexactness.

**Keywords** Nonsmooth optimization, Proximal-projection bundle method, On-demand accuracy, Global convergence

**Mathematics subject classifications (2010)** 90C25, 49J52

**DOI:** 10.19139/soic.v7i1.680

## 1. Introduction

In this paper, we focus on solving problems of the form

$$f_* := \min_{u \in C} f(u), \quad (1.1)$$

where  $C \subseteq \mathbb{R}^n$  is a nonempty closed convex set, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a convex function but not necessarily differentiable.

It is well known that bundle methods are among the most efficient methods for solving nonsmooth optimization problems. For the case where an exact oracle is available, i.e., there is a subroutine that can exactly (in theory) evaluate the function value  $f(u)$  and one arbitrary subgradient  $g(u) \in \partial f(u)$  at any point  $u$ , bundle methods are well studied [1, 2]. However, for some practical problems, such as minimax problems, generalized assignment problems and two-stage stochastic programming problems, etc. (see, e.g. [3, 4]), the exact computation of the function values and subgradients is difficult. In order to solve such kind of problems, a class of bundle methods based on inexact oracle information is proposed [4–14]. In particular, Kiwiel [4] proposed a bundle method with a partially inexact oracle which becomes exact when an objective target level for a descent step is reached, and applied it to solve generalized assignment problems. Oliveira et al. [5] proposed inexact bundle methods for solving two-stage stochastic programming. Fábían [6] presented an asymptotically exact level bundle method that extends the exact version in [15]. Kiwiel [7] proposed a proximal-projection bundle method for constrained problem (1.1), in which a fixed error tolerance of inexactness is used. At each iteration of the algorithm in [7], two subproblems are

---

\*Correspondence to: Chunming Tang (Email: cmtang@gxu.edu.cn). College of Mathematics and Information Science, Guangxi University, No.100, Daxue East Road, Nanning, Guangxi, P. R. China (530004).

solved: a proximal subproblem which is used to generate a linear model of the objective function, and a projection subproblem which is to obtain a trial point.

Recently, a class of so-called bundle methods for *oracles with on-demand accuracy* has been proposed [3, 8, 16]. Particularly, Oliveira and Sagastizábal [3] developed level bundle methods for oracles with on-demand accuracy for solving problems of the form (1.1). The accuracy of the oracles is regulated by two parameters: a descent target and an error bound. If the function estimation reaches the descent target, then the corresponding error is bounded by the second parameter. The most interesting feature of oracles with on-demand accuracy is that it covers various of oracles discussed before. Ackooij and Oliveira [8] generalized the on-demand accuracy approach to handle inequality constrained optimization problems. Wolf et al. [16] presented a computational study for oracles of on-demand accuracy in solving two-stage stochastic linear programming problems.

In this paper, by combining the on-demand accuracy approach of [3] with the proximal-projection bundle method of [7], we proposed a proximal-projection bundle method with on-demand accuracy oracles for solving problem (1.1). Our method essentially generalizes the work of Kiwiel [7] from exact and inexact oracles to various oracles, including exact, inexact, partially inexact, asymptotically exact and partially asymptotically exact oracles. At each iteration, a proximal subproblem is solved to generate a linear model of the objective function, and then a projection subproblem is solved to obtain a trial point. Finally, global convergence of the algorithm is established under different types of inexactness.

This paper is organized as follows. In section 2, we recall the condition of oracles with on-demand accuracy and present an example of two-stage stochastic linear programs which satisfies the oracles. In section 3, we present our proximal-projection bundle method for oracles with on-demand accuracy and show some properties of the algorithm. In section 4, we establish global convergence of the algorithm under different types of inexactness. The notations are standard. The Euclidean inner product in  $\mathbb{R}^n$  is denoted by  $\langle x, y \rangle := x^T y$ , and the associated norm by  $\| \cdot \|$ . The subdifferential of  $f$  at  $u \in \mathbb{R}^n$  is denoted by  $\partial f(u) := \{g : f(y) \geq f(u) + \langle g, y - u \rangle, \forall y \in \mathbb{R}^n\}$ , and each element  $g \in \partial f(u)$  is called a subgradient.

## 2. Preliminaries

The oracle with on-demand accuracy proposed by Oliveira and Sagastizábal [3] is described as follows. For given  $u \in C$ , a descent target  $\gamma_u$  and an error bound  $\varepsilon_u \geq 0$ , the approximate function value  $f_u (\approx f(u))$  and the approximate subgradient  $g_u (\approx g(u))$  satisfy the following condition:

$$\begin{cases} f_u = f(u) - \eta(\gamma_u) \text{ with unknown } \eta(\gamma_u) \geq 0, \\ g_u \in \partial_{\eta(\gamma_u)} f(u), \text{ and} \\ \text{whenever } f_u \leq \gamma_u \text{ (descent target reached), the relation } \eta(\gamma_u) \leq \varepsilon_u \text{ holds.} \end{cases} \tag{2.1}$$

From the above relations, when the descent target is reached, the exact function value satisfies the following relation:

$$f(u) \in [f_u, f_u + \varepsilon_u], \text{ whenever } f_u \leq \gamma_u.$$

By suitably choosing the parameters  $\gamma_u$  and  $\varepsilon_u$ , the oracle (2.1) covers various oracles:

- *Exact Oracle*: Set  $\gamma_u = +\infty$  and  $\varepsilon_u = 0$ .
- *Partially Inexact Oracle* [4]: Set  $\gamma_u < +\infty$  and  $\varepsilon_u = 0$ .
- *Inexact Oracle* [5, 7, 10–12]: Set  $\gamma_u = +\infty$  and  $\varepsilon_u \equiv \varepsilon > 0$  (possibly unknown).
- *Asymptotically Exact Oracle* [7, 13, 14]: Set  $\gamma_u = +\infty$  and  $\varepsilon_u \rightarrow 0$  along the iterative process.
- *Partially Asymptotically Exact Oracle* [3]: Set  $\gamma_u < +\infty$  and  $\varepsilon_u \rightarrow 0$ .

As in [3], an additional assumption is needed: there exists a positive constant  $\bar{\eta}$  such that  $\eta(\gamma_u) \leq \bar{\eta}, \forall u \in C$ .

We now provide one example coming from stochastic optimization that is suitable to apply the oracle (2.1).

*Example 1 (On-demand accuracy oracles for stochastic programming [3])* Consider two-stage stochastic linear programming problems [5, 17] with fixed recourse. By discretizing the uncertainty into  $N$  scenarios, we obtain the

form of problem (1.1)

$$\begin{aligned} \min \quad & f(u) = \langle c, u \rangle + \sum_{i=1}^N p_i V_i(u) \\ \text{s.t.} \quad & u \in C = \{u \in \mathbb{R}_+^n : Au = b\}, \end{aligned}$$

where  $u$  is the first-stage decision variable,  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m_1 \times n}$ , and  $b \in \mathbb{R}^{m_1}$ . In addition, the recourse function is

$$V_i(u) = \min_{\pi \in \mathbb{R}_+^{n_2}} \{\langle q, \pi \rangle : W\pi = h_i - T_i u\},$$

corresponding to the  $i$ th scenario  $(h_i, T_i)$ , with probability  $p_i > 0$  for  $h_i \in \mathbb{R}^{m_2}$  and  $T_i \in \mathbb{R}^{m_2 \times n}$ . Here  $\pi$  is the second-stage decision variable.

The above recourse function can be written as its dual form:

$$V_i(u) = \max_{y \in \mathbb{R}^{m_2}} \langle h_i - T_i u, y \rangle \quad \text{s.t.} \quad W^T y \leq q,$$

where  $q \in \mathbb{R}^{n_2}$  and  $W \in \mathbb{R}^{m_2 \times n_2}$ . By solving this linear programming to return a solution with precision up to a given tolerance, one can establish an inexact oracle in the form (2.1), see [3] for more detailed description.

### 3. The proximal-projection bundle method for oracles with on-demand accuracy

In this section, we present our proximal-projection bundle method for oracles with on-demand accuracy to solve problem (1.1). Firstly, we know that problem (1.1) is equivalent to the unconstrained problem

$$\min \quad f_C(u) := f(u) + i_C(u), \quad (3.1)$$

where  $i_C$  is the indicator function of  $C$ , i.e.,  $i_C(u) = 0$  if  $u \in C$ ;  $\infty$  otherwise. Let  $k$  be the current iteration index,  $\{u^j\}_{j=1}^k \subset C$  be a sequence of trial points, and the corresponding approximate values  $f_{u^j}/g_{u^j}$  be produced by the oracle (2.1). For simplicity, denote  $f_u^j := f_{u^j}$ ,  $g_u^j := g_{u^j}$ ,  $\varepsilon_u^j := \varepsilon_{u^j}$  and  $\gamma_u^j := \gamma_{u^j}$ , then the approximate linearizations of  $f$  at  $u^j$  are given by

$$f_j(\cdot) = f_u^j + \langle g_u^j, \cdot - u^j \rangle.$$

In addition, from (2.1) we conclude that

$$\begin{aligned} f(\cdot) &\geq f(u^j) + \langle g_u^j, \cdot - u^j \rangle - \eta(\gamma_u^j) = f_u^j + \langle g_u^j, \cdot - u^j \rangle, \\ f(u^j) &\in [f_u^j, f_u^j + \varepsilon_u^j], \quad \text{whenever} \quad f_u^j \leq \gamma_u^j. \end{aligned} \quad (3.2)$$

Thus, a simple form of the approximate cutting-planes model of  $f$  at the  $k$ th iteration can be defined by

$$\check{f}_k(\cdot) := \max_{j \in J_k} \{f_u^j + \langle g_u^j, \cdot - u^j \rangle\} \leq f(\cdot), \quad (3.3)$$

where  $J_k \subseteq \{1, \dots, k\}$  is some index set. Note that, in what follows, the choice of the model function  $\check{f}_k$  may be different from the form of (3.3), since a subgradient aggregation strategy is adopted.

Based on the idea of proximal bundle methods (see, e.g. [1]) for solving problem (3.1), one may solve the following subproblem to obtain a new trial  $u^{k+1}$ :

$$\min \quad \check{f}_k(\cdot) + i_C(\cdot) + \frac{1}{2t_k} \|\cdot - \hat{u}^k\|^2, \quad (3.4)$$

where  $t_k > 0$  is a stepsize that controls the size of  $\|u^{k+1} - \hat{u}^k\|$ , and  $\hat{u}^k$  (called *stability center*) is the ‘‘best’’ point obtained so far. Usually,  $\hat{u}^k = u^{k(l)}$  for some  $k(l) \leq k$  satisfying  $f_u^{k(l)} = \min_{j=1, \dots, k} f_u^j$ , so from the oracle (2.1) it naturally follows

$$f(\hat{u}^k) \in [f_{\hat{u}}^k, f_{\hat{u}}^k + \varepsilon_{\hat{u}}^{k(l)}], \quad \text{whenever} \quad f_{\hat{u}}^k \leq \gamma_{\hat{u}}^k. \quad (3.5)$$

However, the subproblem (3.4) is usually not easy to solve, so by making use the proximal-projection idea of Kiwiel [7], we solve two easier subproblems instead. One is an unconstrained proximal subproblem which is used to generate an aggregate linearization of  $\check{f}$ , and the other subproblem based on this linearization is solved to produce a new trial point. The second subproblem is equivalent to projecting a certain point onto the feasible set  $C$ , which can have a closed-form solution if  $C$  has some special structure.

Now, we present the details of our algorithm, which is a generalized version of that in [7]. The main difference lies in Step 6 which incorporates the strategy of the on-demand accuracy oracle (2.1).

*Algorithm 3.1*

**Step 0** (Initialization). Select  $u^1 \in C$ ,  $\kappa \in (0, 1)$ ,  $t_{\min} > 0$ , a stepsize  $t_1 \geq t_{\min}$ . Call the oracle (2.1) at  $u^1$  to compute the corresponding  $f_u^1/g_u^1$  information. Choose an error bound  $\varepsilon_u^1 \geq 0$  and a descent target  $\gamma_u^1 = +\infty$ . Set  $\hat{u}^1 := u^1$ ,  $f_{\hat{u}}^1 := f_u^1$ ,  $f_0 := f_1 = f_u^1 + \langle g_u^1, \cdot - u^1 \rangle$ ,  $\bar{v}_C^0 := \langle p_C^0, \cdot - u^1 \rangle$  with  $p_C^0 = 0$ ,  $i_t^1 := 0$ ,  $l = 1$ ,  $k(l) = 1$  and  $k := 1$ .

**Step 1** (Model function). Choose  $\check{f}_k : \mathbb{R}^n \rightarrow \mathbb{R}$  closed convex and such that

$$\max\{\bar{f}_{k-1}, f_k\} \leq \check{f}_k \leq f.$$

**Step 2** (Proximal subproblem). Set

$$\check{u}^{k+1} := \arg \min \left\{ \phi_f^k(\cdot) := \check{f}_k(\cdot) + \bar{v}_C^{k-1}(\cdot) + \frac{1}{2t_k} \|\cdot - \hat{u}^k\|^2 \right\}, \tag{3.6}$$

$$\bar{f}_k(\cdot) := \check{f}_k(\check{u}^{k+1}) + \langle p_f^k, \cdot - \check{u}^{k+1} \rangle \text{ with } p_f^k := \frac{1}{t_k}(\hat{u}^k - \check{u}^{k+1}) - p_C^{k-1}. \tag{3.7}$$

**Step 3** (Projection subproblem). Set

$$u^{k+1} := \arg \min \left\{ \phi_C^k(\cdot) := \bar{f}_k(\cdot) + i_C(\cdot) + \frac{1}{2t_k} \|\cdot - \hat{u}^k\|^2 \right\}, \tag{3.8}$$

$$\bar{v}_C^k(\cdot) := \langle p_C^k, \cdot - u^{k+1} \rangle \text{ with } p_C^k := \frac{1}{t_k}(\hat{u}^k - u^{k+1}) - p_f^k, \tag{3.9}$$

$$v_k := f_{\hat{u}}^k - \bar{f}_k(u^{k+1}), p^k := \frac{1}{t_k}(\hat{u}^k - u^{k+1}), \text{ and } \epsilon_k := v_k - t_k \|p^k\|^2. \tag{3.10}$$

**Step 4** (Stopping rule). If  $\max\{\|p^k\|, \epsilon_k\} = 0$ , stop.

**Step 5** (Noise attenuation). If  $v_k < -\epsilon_k$ , set  $t_k := 10t_k$ ,  $i_t^k := k$ , and go back to Step 2.

**Step 6** (Calling oracle). Select a new error bound  $\varepsilon_u^{k+1} \geq 0$  and a new descent target  $\gamma_u^{k+1} \in R \cup \{+\infty\}$ . Call the oracle (2.1) to obtain  $f_u^{k+1}$  and  $g_u^{k+1}$ .

**Step 7** (Descent test). If

$$f_u^{k+1} \leq f_{\hat{u}}^k - \kappa v_k, \tag{3.11}$$

set  $\hat{u}^{k+1} := u^{k+1}$ ,  $f_{\hat{u}}^{k+1} := f_u^{k+1}$ ,  $i_t^{k+1} := 0$ ,  $k(l+1) := k+1$ , and increase  $l$  by 1 (descent step); otherwise, set  $\hat{u}^{k+1} := \hat{u}^k$ ,  $f_{\hat{u}}^{k+1} := f_{\hat{u}}^k$ , and  $i_t^{k+1} := i_t^k$  (null step).

**Step 8** (Stepsize updating). If  $k(l+1) = k+1$ , select  $t_{k+1} \geq t_k$ . Otherwise, either set  $t_{k+1} := t_k$  or select  $t_{k+1} \in [t_{\min}, t_k]$  if  $i_t^{k+1} = 0$ .

**Step 9** (Loop). Set  $k := k+1$ , and go to Step 1.

Some comments about the algorithm are given below.

*Remark 1*

(i) The choice of the model function  $\check{f}_k$  is very flexible. The simplest choice  $\check{f}_k = \max\{\bar{f}_{k-1}, f_k\}$  only contains two linear functions, but for numerical stability, some other linearizations should be included.

(ii) Solving two subproblems at Steps 2 and 3 can be viewed as the alternating linearization method (e.g. [18]) being applied to problem (3.4).

(iii) Solving subproblem (3.8) is equivalent to projecting the point  $\hat{u}^k - t_k p_f^k$  onto the feasible set  $C$ .

(iv) Inexactness is discovered via  $v_k < -\epsilon_k$  at Step 5, and the stepsize  $t_k$  is increased until  $v_k \geq -\epsilon_k$  is generated.

(v) The descent target and the error bound are updated at Step 6, and the detailed rules to ensure convergence of the algorithm are given in the next section.

Denote the optimality measure

$$V_k := \max\{\|p^k\|, \epsilon_k + \langle p^k, \hat{u}^k \rangle\}. \quad (3.12)$$

When  $\{\hat{u}^k\}$  is bounded and  $\varepsilon_u^{k(l)}$  is not updated in (3.22) increasing  $t_k$  drives  $V_k$  to 0, one can obtain  $f(\hat{u}^k) \leq f_* + \varepsilon_u^{k(l)}$  asymptotically.

The following lemma states some important properties of Algorithm 3.1, in which most of the results are borrowed from [7], but for completeness, we present the whole proof.

*Lemma 3.2*

(i) The vectors  $p_f^k$  and  $p_C^k$  defined in (3.7) and (3.9) satisfy

$$p_f^k \in \partial \check{f}_k(\check{u}^{k+1}) \quad \text{and} \quad p_C^k \in \partial i_C(u^{k+1}). \quad (3.13)$$

The linearizations  $\bar{f}_k$ ,  $\bar{v}_C^k$  and  $\bar{f}_C^k$  satisfy the following inequalities

$$\bar{f}_k \leq \check{f}_k, \quad \bar{v}_C^k \leq i_C \quad \text{and} \quad \bar{f}_C^k := \bar{f}_k + \bar{v}_C^k \leq f_C. \quad (3.14)$$

(ii) The aggregate subgradient  $p_k$  of (3.10) and the above linearization  $\bar{f}_C^k$  can be expressed as follows

$$p^k = p_f^k + p_C^k = \frac{1}{t_k}(\hat{u}^k - u^{k+1}),$$

$$\bar{f}_C^k(\cdot) = \bar{f}_k(u^{k+1}) + \langle p^k, \cdot - u^{k+1} \rangle.$$

(iii) The predicted descent  $v_k$  and the aggregate linearization error  $\epsilon_k$  of (3.10) satisfy

$$v_k = t_k \|p^k\|^2 + \epsilon_k \quad \text{and} \quad \epsilon_k = f_{\hat{u}}^k - \bar{f}_C^k(\hat{u}^k). \quad (3.15)$$

(iv) The aggregate linearization  $\bar{f}_C^k$  satisfies

$$f_{\hat{u}}^k - \epsilon_k + \langle p^k, \cdot - \hat{u}^k \rangle = \bar{f}_C^k(\cdot) \leq f_C(\cdot). \quad (3.16)$$

(v) The optimality measure  $V_k$  of (3.12) satisfies

$$V_k \leq \max\{\|p^k\|, \epsilon_k\}(1 + \|\hat{u}^k\|) \quad (3.17)$$

and

$$f_{\hat{u}}^k \leq f_C(u) + V_k(1 + \|u\|), \quad \forall u. \quad (3.18)$$

(vi) The following relations hold:

$$v_k \geq -\epsilon_k \Leftrightarrow t_k \|p^k\|^2 / 2 \geq -\epsilon_k \Leftrightarrow v_k \geq t_k \|p^k\|^2 / 2, \quad v_k \geq \epsilon_k. \quad (3.19)$$

Moreover, if  $f_{\hat{u}}^k \leq \gamma_{\hat{u}}^k$ , we have  $-\epsilon_k \leq \varepsilon_u^{k(l)}$  and

$$v_k \geq \max \left\{ \frac{t_k \|p^k\|^2}{2}, |\epsilon_k| \right\} \quad \text{if } v_k \geq -\epsilon_k, \quad (3.20)$$

$$V_k \leq \max \left\{ \left( \frac{2v_k}{t_k} \right)^{1/2}, v_k \right\} (1 + \|\hat{u}^k\|) \quad \text{if } v_k \geq -\epsilon_k, \quad (3.21)$$

$$V_k < \left( \frac{2\varepsilon_u^{k(l)}}{t_k} \right)^{1/2} (1 + \|\hat{u}^k\|) \quad \text{if } v_k < -\epsilon_k. \quad (3.22)$$

*Proof*

(i) From the optimality condition of (3.6) and the fact that  $\nabla \bar{v}_C^{k-1} = p_C^{k-1}$ , we have

$$0 \in \partial \phi_f^k(\check{u}^{k+1}) = \partial \check{f}_k(\check{u}^{k+1}) + p_C^{k-1} + \frac{1}{t_k}(\check{u}^{k+1} - \hat{u}^k) = \partial \check{f}_k(\check{u}^{k+1}) - p_f^k,$$

which implies  $p_f^k \in \partial \check{f}_k(\check{u}^{k+1})$ . Furthermore, the relation  $\bar{f}_k \leq \check{f}_k$  follows from  $\bar{f}_k(\check{u}^{k+1}) = \check{f}_k(\check{u}^{k+1})$ . Similarly, from the optimality condition of (3.8), we have  $p_C^k \in \partial i_C(u^{k+1})$  and  $\bar{v}_C^k \leq i_C$ . Hence, (3.14) holds.

(ii) By (3.9), we obtain

$$p_f^k + p_C^k = p_f^k + \frac{1}{t_k}(\hat{u}^k - u^{k+1}) - p_f^k = \frac{1}{t_k}(\hat{u}^k - u^{k+1}) = p^k.$$

Using the linearity of  $\bar{f}_C^k(\cdot)$  and (3.7), we derive

$$\begin{aligned} \bar{f}_C^k(\cdot) &= \bar{f}_k(\cdot) + \bar{v}_C^k(\cdot) \\ &= \check{f}_k(\check{u}^{k+1}) + \langle p_f^k, \cdot - \check{u}^{k+1} \rangle + \langle p_C^k, \cdot - u^{k+1} \rangle \\ &= \bar{f}_k(u^{k+1}) - \langle p_f^k, u^{k+1} - \check{u}^{k+1} \rangle + \langle p_f^k, \cdot - \check{u}^{k+1} \rangle + \langle p_C^k, \cdot - u^{k+1} \rangle \\ &= \bar{f}_k(u^{k+1}) + \langle p_f^k, \cdot - u^{k+1} \rangle + \langle p_C^k, \cdot - u^{k+1} \rangle \\ &= \bar{f}_k(u^{k+1}) + \langle p^k, \cdot - u^{k+1} \rangle. \end{aligned}$$

(iii) Combining (3.10) and (ii), we have

$$\begin{aligned} \epsilon_k = v_k - t_k \|p^k\|^2 &= f_{\hat{u}}^k - \bar{f}_k(u^{k+1}) - t_k \|p^k\|^2 \\ &= f_{\hat{u}}^k - \bar{f}_C^k(\hat{u}^k) + \langle p^k, \hat{u}^k - u^{k+1} \rangle - t_k \|p^k\|^2 \\ &= f_{\hat{u}}^k - \bar{f}_C^k(\hat{u}^k). \end{aligned}$$

(iv) Since  $\epsilon_k = v_k - t_k \|p^k\|^2 = f_{\hat{u}}^k - \bar{f}_k(u^{k+1}) - t_k \|p^k\|^2$ , the aggregate lineaization satisfies

$$f_{\hat{u}}^k - \epsilon_k + \langle p^k, \cdot - \hat{u}^k \rangle = \bar{f}_k(u^{k+1}) + \langle p^k, \cdot - u^{k+1} \rangle = \bar{f}_C^k(\cdot) \leq f_C(\cdot).$$

(v) By the Cauchy-Schwarz inequality and (3.12), we obtain

$$\begin{aligned} V_k = \max\{\|p^k\|, \epsilon_k + \langle p^k, \hat{u}^k \rangle\} &\leq \max\{\|p^k\|, \epsilon_k + \|p^k\| \|\hat{u}^k\|\} \\ &\leq \max\{\|p^k\|, \epsilon_k\} + \|p^k\| \|\hat{u}^k\| \\ &\leq \max\{\|p^k\|, \epsilon_k\} (1 + \|\hat{u}^k\|). \end{aligned}$$

From (3.16), we have  $f_{\hat{u}}^k \leq f_C(u) + \epsilon_k - \langle p^k, u - \hat{u}^k \rangle = f_C(u) + \epsilon_k - \langle p^k, u \rangle + \langle p^k, \hat{u}^k \rangle$  for all  $u$ , and

$$\begin{aligned} \epsilon_k - \langle p^k, u \rangle + \langle p^k, \hat{u}^k \rangle &\leq \|p^k\| \|u\| + \epsilon_k + \langle p^k, \hat{u}^k \rangle \\ &\leq \max\{\|p^k\|, \epsilon_k + \langle p^k, \hat{u}^k \rangle\} (1 + \|u\|) \\ &= V_k (1 + \|u\|). \end{aligned}$$

Thus  $f_{\hat{u}}^k \leq f_C(u) + V_k (1 + \|u\|)$  for all  $u$  holds.

(vi) From part (iii) above, (3.19) holds immediately. Furthermore, by (3.15), (3.14) and (3.5) with  $f_C(\hat{u}^k) = f(\hat{u}^k)$  ( $\hat{u}^k \in C$ ), we conclude that

$$-\epsilon_k = \bar{f}_C^k(\hat{u}^k) - f_{\hat{u}}^k \leq f_C(\hat{u}^k) - f_{\hat{u}}^k = f(\hat{u}^k) - f_{\hat{u}}^k \leq \epsilon_u^{k(l)}.$$

Combining  $v_k \geq -\epsilon_k$ ,  $v_k \geq \epsilon_k$  and  $v_k \geq t_k \|p^k\|^2/2$ , we have (3.20). By (3.17),  $\|p^k\| \leq (\frac{2v_k}{t_k})^{1/2}$ , and  $\epsilon_k \leq v_k$ , we obtain (3.21). Finally, from the relations  $\|p^k\|^2 < \frac{-2\epsilon_k}{t_k}$ , and  $-\epsilon_k \leq \epsilon_u^{k(l)}$ , we obtain (3.22).  $\square$

#### 4. Global convergence

In this section, we establish the global convergence of Algorithm 3.1 under different types of oracles. The oracles depend on two parameters: the error bound  $\varepsilon_u$  and the descent target  $\gamma_u$ . Table 1 provides the choice of the parameters for various instances described in Section 2, including Exact (Ex), Partially Inexact (PI), Inexact(IE), Asymptotically Exact (AE) and Partially Asymptotically Exact (PAE) oracles. The constants are selected as  $\theta, \kappa \in (0, 1)$ , and  $\kappa_\varepsilon \in (0, \kappa)$ .

Table 1. The choices for the error bound and the descent target

Instances	$\varepsilon_u^{k+1}$	$\gamma_u^{k+1}$
Ex	0	$+\infty$
PI	0	$f_{\hat{u}}^k - \theta\kappa v_k$
IE	$\varepsilon > 0$	$+\infty$
AE	$\kappa_\varepsilon v^k$	$+\infty$
PAE	$\kappa_\varepsilon v^k$	$f_{\hat{u}}^k - \theta\kappa v_k$

##### Lemma 4.1

The descent target is always reached at the stability centers, i.e.,  $f_{\hat{u}}^k \leq \gamma_{\hat{u}}^k$  for all  $k \geq 1$ .

##### Proof

The claim holds obviously for instances Ex, IE and AE, since  $\gamma_u^k = +\infty$ .

Next we analyze instances PI and PAE.

For  $k = 1$ , from Step 0 we know that  $\hat{u}^1 := u^1$ ,  $f_{\hat{u}}^1 := f_u^1$ , and  $\gamma_{\hat{u}}^1 := \gamma_u^1 = +\infty$ , thus  $f_{\hat{u}}^1 \leq \gamma_{\hat{u}}^1$  holds.

For  $k > 1$ , since  $\theta \in (0, 1)$ , once the descent test (3.11) is satisfied at iteration  $k - 1$ , we have

$$f_{\hat{u}}^k \leq f_{\hat{u}}^{k-1} - \kappa v_{k-1} \leq f_{\hat{u}}^{k-1} - \theta\kappa v_{k-1} = \gamma_{\hat{u}}^k = \gamma_{\hat{u}}^k.$$

□

##### Lemma 4.2

If either Algorithm 3.1 terminates at Step 4 with iteration  $k$ , or the number of loops between Steps 2 and 5 is infinite, then

- (i)  $\hat{u}^k$  is an optimal solution to problem (1.1) for instances Ex and PI.
- (ii)  $\hat{u}^k$  is  $\varepsilon$ -optimal, i.e.,  $f(\hat{u}^k) \leq f_* + \varepsilon$ , for instance IE.
- (iii)  $\hat{u}^k$  is  $\varepsilon_u^{k(l)}$ -optimal, i.e.,  $f(\hat{u}^k) \leq f_* + \varepsilon_u^{k(l)}$ , for instances AE and PAE.

##### Proof

For the case where Algorithm 3.1 terminates at Step 4. From (3.17), it follows that  $V_k = 0$ . Thus, we have  $f_{\hat{u}}^k \leq \inf f_C = f_*$  by (3.18). From (3.2), we have (i)  $f(\hat{u}^k) = f_{\hat{u}}^k \leq f_*$  for instances Ex and PI; (ii)  $f(\hat{u}^k) \leq f_* + \varepsilon$  for instance IE; (iii)  $f(\hat{u}^k) \leq f_* + \varepsilon_u^{k(l)}$  for instances AE and PAE.

For the case where the loop between Steps 2 and 5 is infinite. From Lemma 4.1 and the condition at Step 5, we obtain (3.22), which in turn implies  $V_k \rightarrow 0$  since  $t_k \uparrow \infty$ . Thus,  $f_{\hat{u}}^k \leq f_C(u)$ ,  $\forall u$  by (3.18), and then  $f_{\hat{u}}^k \leq \inf_{u \in \mathbb{R}^n} f_C(u) = f_*$ . So from (3.2) again, we can obtain the claims by repeating the lines in first case. □

From Lemma 4.2 above, we now assume that Algorithm 3.1 neither terminates nor loops infinitely between Steps 2 and 5. As in [7], it is assumed that the model subgradients  $p_f^k \in \partial f_k(\check{u}^{k+1})$  in (3.13) satisfy that  $\{p_f^k\}$  is bounded if  $\{\check{u}^k\}$  is bounded.

The following analysis is divided into two cases: finitely many descent steps and infinitely many descent steps. We consider the first case, which involves two subcases:  $t_\infty := \lim_k t_k = \infty$  and  $t_\infty < \infty$ . We first analyze the case of  $t_\infty = \infty$ .

*Lemma 4.3*

Suppose that there exists a last descent index  $\bar{k}$  such that only null steps occur for all  $k \geq \bar{k}$ , and  $t_\infty = \infty$ . Let  $\mathcal{K} := \{k \geq \bar{k} : t_{k+1} > t_k\}$ , then  $V_k \xrightarrow{\mathcal{K}} 0$  at Step 5.

*Proof*

For the last time  $t_k$  increases before Step 5 for  $k \in \mathcal{K}$ , we have  $V_k < (\frac{2\varepsilon_{\hat{u}^k}}{t_k})^{1/2}(1 + \|\hat{u}^k\|)$ , thus  $t_k \rightarrow \infty$  gives  $V_k \xrightarrow{\mathcal{K}} 0$ . □

Next, we consider the case where  $t_\infty < \infty$ .

*Lemma 4.4*

Suppose that there exists  $\bar{k}$  such that  $\hat{u}^k = \hat{u}^{\bar{k}}$  and  $t_{\min} \leq t_{k+1} \leq t_k$  for all  $k \geq \bar{k}$ . If the descent criterion (3.11) fails for all  $k \geq \bar{k}$ , then  $V_k \rightarrow 0$ .

*Proof*

For  $k \geq \bar{k}$ , combining  $t_k \geq t_{\min}$ ,  $\hat{u}^k = \hat{u}^{\bar{k}}$  and the proof of [7, Lemma 3.2], we have  $V_k \rightarrow 0$ . □

We finish the case of finitely many descent steps. The following lemma comes from [7, Lemma 3.3].

*Lemma 4.5*

Suppose that there exists  $\bar{k}$  such that only null steps occur for all  $k \geq \bar{k}$ . Let  $\mathcal{K} := \{k \geq \bar{k} : t_{k+1} > t_k\}$  if  $t_k \rightarrow \infty$ ;  $\mathcal{K} := \{k : k \geq \bar{k}\}$  otherwise. Then  $V_k \xrightarrow{\mathcal{K}} 0$ .

*Theorem 4.6*

Suppose that finitely many descent steps occur and let  $\hat{u}^{\bar{k}}$  be the last stability center. Then  $\hat{u}^{\bar{k}}$  is an  $\varepsilon_{\hat{u}^{\bar{k}}}$ -optimal solution to problem (1.1).

*Proof*

We have  $\hat{u}^k = \hat{u}^{\bar{k}}$ ,  $f_{\hat{u}}^k = f_{\hat{u}}^{\bar{k}}$  for all  $k \geq \bar{k}$ . By (3.18) and lemma 4.5, we obtain  $f(\hat{u}^{\bar{k}}) \leq f_* + \varepsilon_{\hat{u}^{\bar{k}}}$ . Thus,  $\hat{u}^{\bar{k}}$  is an  $\varepsilon_{\hat{u}^{\bar{k}}}$ -optimal solution to problem (1.1). □

*Remark 2*

From Table 1, Theorem 4.6 shows that  $\hat{u}^{\bar{k}}$  is an optimal solution for instances Ex and PI; an  $\varepsilon$ -optimal solution for IE; and an  $\varepsilon_{\hat{u}^{\bar{k}}}$ -optimal solution for AE and PAE.

We now analyze the case of infinitely many descent steps. The following lemma is borrowed from [7, Lemma 3.4].

*Lemma 4.7*

Suppose that infinitely many descent steps occur and  $f_{\hat{u}}^\infty := \lim_k f_{\hat{u}}^k > -\infty$ . Let  $K := \{k : f_{\hat{u}}^{k+1} < f_{\hat{u}}^k\}$ . Then  $v_k \xrightarrow{K} 0$  and  $\underline{\lim}_{k \in K} V_k = 0$ . Moreover, if  $\{\hat{u}^k\}$  is bounded, then  $V_k \xrightarrow{K} 0$ .

*Theorem 4.8*

- (i)  $f_* \leq \underline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \overline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq f_{\hat{u}}^\infty + \varepsilon$  for instance IE in Table 1.
- (ii)  $f_* \leq \underline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \overline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq f_{\hat{u}}^\infty$  for the remaining instances in Table 1.
- (iii) If  $f_* > -\infty$ , then  $\underline{\lim}_{k \in K} V_k = 0$  and  $f_{\hat{u}}^k \downarrow f_{\hat{u}}^\infty \leq f_*$ .

*Proof*

Since  $\{\hat{u}^{k+1}\} \subset C$ , we have  $f_* := \inf_C f \leq f(\hat{u}^{k+1})$ , thus

$$f_* \leq \underline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \overline{\lim}_{k \in K} f(\hat{u}^{k+1}). \tag{4.1}$$

(i) For instance IE, we have  $\varepsilon_{\hat{u}^{k+1}} = \varepsilon$  and  $f_{\hat{u}^{k+1}} \leq \gamma_{\hat{u}^{k+1}} = \infty$  for all  $k \in K$ , thus  $f(\hat{u}^{k+1}) \leq f_{\hat{u}^{k+1}} + \varepsilon, \forall k \in K$  and therefore

$$\overline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \lim_{k \in K} f_{\hat{u}^{k+1}} + \varepsilon = f_{\hat{u}}^\infty + \varepsilon.$$



This together with (4.1) shows claim (i).

(ii) We split our analysis for other instances in Table 1:

- Instance Ex has  $\varepsilon_{\hat{u}}^{k+1} = 0$  and  $f_{\hat{u}}^{k+1} \leq \gamma_{\hat{u}}^{k+1} = \infty$ , then  $f(\hat{u}^{k+1}) = f_{\hat{u}}^{k+1}$  by (3.2). Hence, it follows

$$\overline{\lim}_{k \in K} f(\hat{u}^{k+1}) = \lim_{k \in K} f_{\hat{u}}^{k+1} = f_{\hat{u}}^{\infty}.$$

- Instance PI has  $\varepsilon_{\hat{u}}^{k+1} = 0$  and  $\gamma_{\hat{u}}^{k+1} = f_{\hat{u}}^k - \theta \kappa v_k$  for  $k \in K$ , then  $f_{\hat{u}}^{k+1} \leq f_{\hat{u}}^k - \kappa v_k \leq f_{\hat{u}}^k - \theta \kappa v_k = \gamma_{\hat{u}}^{k+1}$ , and  $f(\hat{u}^{k+1}) = f_{\hat{u}}^{k+1}$  by (3.2). Hence

$$\overline{\lim}_{k \in K} f(\hat{u}^{k+1}) = \lim_{k \in K} f_{\hat{u}}^{k+1} = f_{\hat{u}}^{\infty}.$$

- Instance AE has  $\varepsilon_{\hat{u}}^{k+1} = \kappa_{\epsilon} v_k$  and  $f_{\hat{u}}^{k+1} \leq \gamma_{\hat{u}}^{k+1} = \infty$  for  $k \in K$ , then

$$f(\hat{u}^{k+1}) \leq f_{\hat{u}}^{k+1} + \varepsilon_{\hat{u}}^{k+1} \leq f_{\hat{u}}^{k+1} + \kappa_{\epsilon} v_k,$$

which together with  $v_k \xrightarrow{K} 0$  (Lemma 4.7) gives

$$\overline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \lim_{k \in K} f_{\hat{u}}^{k+1} = f_{\hat{u}}^{\infty}.$$

- Instance PAE has  $\varepsilon_{\hat{u}}^{k+1} = \kappa_{\epsilon} v_k$  and  $\gamma_{\hat{u}}^{k+1} = f_{\hat{u}}^k - \theta \kappa v_k$  for  $k \in K$ , then

$$f_{\hat{u}}^{k+1} \leq f_{\hat{u}}^k - \kappa v_k \leq f_{\hat{u}}^k - \theta \kappa v_k = \gamma_{\hat{u}}^{k+1},$$

so  $f(\hat{u}^{k+1}) \leq f_{\hat{u}}^{k+1} + \kappa_{\epsilon} v_k$  by (3.2). Similarly, we have  $v_k \xrightarrow{K} 0$  by Lemma 4.7, therefore

$$\overline{\lim}_{k \in K} f(\hat{u}^{k+1}) \leq \lim_{k \in K} f_{\hat{u}}^{k+1} = f_{\hat{u}}^{\infty}.$$

(iii) If  $f_* > -\infty$ , then  $f_{\hat{u}}^{\infty} > -\infty$  by (i) and (ii) above. So by Lemma 4.7, we have  $\underline{\lim}_{k \in K} V_k = 0$ . Passing to the limit in (3.18), we obtain  $f_{\hat{u}}^{\infty} \leq \inf f_C = f_*$ .  $\square$

## 5. Conclusions

In this paper, we have presented a proximal-projection bundle method for oracles with on-demand accuracy for minimizing a convex function over a closed convex set. Our method extends the inexact oracle of [7] to oracles with on-demand accuracy. By suitably choosing the parameters of the oracle, the descent target is always reached at the stability centers, which is the key to establish global convergence of the algorithm under different types of inexactness.

## Acknowledgement

This work was supported by the National Natural Science Foundation (11761013, 71861002), and Guangxi Natural Science Foundation (2017GXNSFBA198238) of China.

## REFERENCES

- [1] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. Sagastizábal, Numerical Optimization: Theoretical and Practical Aspects. Second ed. Springer-Verlag, Berlin Heidelberg New York (2006).
- [2] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer science & business media, vol. 305, 2013.

- [3] W. Oliveira and C. Sagastizábal, *Level bundle methods for oracles with on demand accuracy*, Optimization Methods and Software, vol. 29, pp. 1180-1209, 2014.
- [4] K.C. Kiwiel, *Bundle methods for convex minimization with partially inexact oracles*, Tech. Rep., Systems Research Institute, Polish Academy of Sciences, Warsaw, 2009.
- [5] W. Oliveira, C. Sagastizábal and S. Scheimberg, *Inexact bundle methods for two-stage stochastic programming*, SIAM Journal on Optimization, vol. 21, pp. 517-544, 2011.
- [6] C. I. Fábíán, *Bundle-type methods for inexact data*, Central European Journal of Operations Research, vol. 8, pp. 35-55, 2000.
- [7] K. C. Kiwiel, *A proximal-projection bundle method for Lagrangian relaxation, including semidefinite programming*, SIAM Journal on Optimization, vol. 17, pp. 1015-1034, 2006.
- [8] W. Ackooij and W. Oliveira, *Level bundle methods for constrained convex optimization with various oracles*, Computational Optimization and Applications, vol. 57, pp. 555-597, 2014.
- [9] K. C. Kiwiel, *An algorithm for nonsmooth convex minimization with errors*, Mathematics of Computation, vol. 45(171), pp. 173-180, 1985.
- [10] K. C. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM Journal on Optimization, vol. 16, pp. 1007-1023, 2006.
- [11] W. Oliveira, C. Sagastizbal and C. Lemarchal, *Convex proximal bundle methods in depth: a unified analysis for inexact oracles*, Mathematical Programming, vol. 148, pp. 241-277, 2014.
- [12] J. Malick, W. Oliveira and S. Zaourar-Michel, *Uncontrolled inexact information within bundle methods*, EURO Journal on Computational Optimization, Springer, vol. 5, pp. 5-29, 2017.
- [13] G. Zakeri, A. B. Philpott and D. M. Ryan, *Inexact cuts in benders decomposition*, SIAM Journal on Optimization, vol. 10, pp. 643-657, 2000.
- [14] K. C. Kiwiel, *Bundle methods for convex minimization with partially inexact oracles*, Technical report, 2009.
- [15] C. Lemaréchal, A. S. Nemirovskii and Y. E. Nesterov, *New variants of bundle methods*, Mathematical Programming, vol. 69, pp. 111-147, 1995.
- [16] C. Wolf, C. I. Fábíán, A. Koberstein, et al. *Applying oracles of on-demand accuracy in two-stage stochastic programming - A computational study*, European Journal of Operational Research, vol. 239, pp. 437-448, 2014.
- [17] A. Shapiro, D. Dentcheva and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, Society for Industrial and Applied Mathematics, 2009.
- [18] K. C. Kiwiel, C. H. Rosa and A. Ruszczyński, *Proximal decomposition via alternating linearization*, SIAM Journal on Optimization, vol. 9, pp. 668-689, 1999.