Computational Optimization and Applications
https://doi.org/10.1007/s10589-020-00208-9

Acceleration techniques for level bundle methods in weakly smooth convex constrained optimization

Yunmei Chen[1] · Xiaojing Ye[2] · Wei Zhang[1]

Received: 9 June 2017

Abstract
We develop a unified level-bundle method, called accelerated constrained level-bundle (ACLB) algorithm, for solving constrained convex optimization problems. where the objective and constraint functions can be nonsmooth, weakly smooth, and/or smooth. ACLB employs Nesterov's accelerated gradient technique, and hence retains the iteration complexity as that of existing bundle-type methods if the objective or one of the constraint functions is nonsmooth. More importantly, ACLB can significantly reduce iteration complexity when the objective and all constraints are (weakly) smooth. In addition, if the objective contains a nonsmooth component which can be written as a specific form of maximum, we show that the iteration complexity of this component can be much lower than that for general nonsmooth objective function. Numerical results demonstrate the effectiveness of the proposed algorithm.

Keywords Convex optimization · Acceleration · Bundle method · Functional constrained optimization

✉ Wei Zhang
weizhang657@ufl.edu

Yunmei Chen
yun@math.ufl.edu

Xiaojing Ye
xye@gsu.edu

[1] Department of Mathematics, University of Florida, Gainesville, FL 32611, USA

[2] Department of Mathematics and Statistics, George State University, Atlanta, GA 30303, USA

Published online: 06 July 2020

1 Introduction

1.1 Problem description

In this paper, we are interested in solving the constrained optimization problem where the objective function F may contain either or both of f_0 and f as follows:

$$\min_{x\in X} \{F(x) := f_0(x) + f(x)\}, \quad \text{s.t.} \quad g_i(x) \le 0, \ \forall\, i = 1, \dots, n_c. \tag{1}$$

Here X is a compact convex set in $\mathbb{R}^{d_X}$, the functions $f_0, g_i : X \to \mathbb{R}$ are proper, closed, and convex functions, and there exist $L_0, L_i > 0$ and $\rho_0, \rho_i \in [0, 1]$ such that

$$f_0(\bar{x}) - f_0(x) - \langle \xi(x), \bar{x} - x\rangle \le \frac{L_0}{1+\rho_0}\|\bar{x} - x\|^{1+\rho_0}, \quad \forall\, x, \bar{x} \in X, \tag{2}$$

$$g_i(\bar{x}) - g_i(x) - \langle \zeta_i(x), \bar{x} - x\rangle \le \frac{L_i}{1+\rho_i}\|\bar{x} - x\|^{1+\rho_i}, \quad \forall\, x, \bar{x} \in X, \quad i = 1, \dots, n_c. \tag{3}$$

In (1), f is a special max-type (possibly nonsmooth) function:

$$f(x) := \max_{y\in Y} \{\langle Ax, y\rangle - \chi(y)\}, \tag{4}$$

where $Y \subseteq \mathbb{R}^{d_Y}$ is a compact convex set, $\chi : Y \to \mathbb{R}$ is a proper, closed convex function, and $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator (such as a matrix). In (2)–(4), $\|\cdot\|$ is the standard Euclidean norm, $\langle \cdot, \cdot \rangle$ is the inner product of vectors in $\mathbb{R}^n$, and $\xi(x) \in \partial f_0(x)$ and $\zeta_i(x) \in \partial g_i(x)$ are any (sub)gradient of f_0 and g_i at x, respectively. The conditions given in (2) and (3) allow us to present our algorithm and convergence results in a unified framework with f_0 and g_i's of any smoothness level $\rho_0, \rho_{g_i} \in [0, 1]$. More precisely, we know that f_0 is nonsmooth if $\rho_0 = 0$, weakly smooth if $\rho_0 \in (0, 1)$, and smooth if $\rho_0 = 1$. Similar statements hold for the constraint functions g_i's. Therefore, for example, if $\rho_0 = 0$ but $\rho_i = 1$ for all $i = 1, \dots, n_c$, then we will be dealing with problem (1) that has nonsmooth objective function F and smooth constraints g_i, etc. Moreover, if $\rho_0 = 1$ (or f_0 is not present), then we can show that the problem with a nonsmooth max-type f given in (4) can be handled with a better iteration complexity than that for treating f as a generic nonsmooth objective function.

The goal of this paper is develop an algorithm that solves the problem (1) uniformly for smooth, weakly smooth and nonsmooth f_0 and g_i, where the (whole or part of) objective function F can be written as the structured, possibly nonsmooth f as in (4). This type of problems has extensive real-world applications in signal/image processing [2, 24], tensor decomposition [15, 26], overlapped group lasso [10, 18], and graph regularization [25], etc. To cope with the special structure of f, we consider a prox-function (also known as a distance generating function) $v : Y \to \mathbb{R}$, which is strongly convex with modulus σ_v. Then the Bregman divergence associated with this v is defined by

$$V(y) := v(y) - v(c_v) - \langle \nabla v(c_v), y - c_v \rangle, \tag{5}$$

where $c_v := \arg\min_{v \in Y} v(y)$. By employing Nesterov's smoothing technique [22], we can approximate $f(\cdot)$ by the smooth function f_η defined as follows,

$$f_\eta(x) := \max_{y \in Y} \{\langle Ax, y \rangle - \chi(y) - \eta V(y)\}, \tag{6}$$

where η is called the smoothing parameter. It is shown in [22] that $f_\eta(x)$ is differentiable and has Lipschitz continuous gradient ∇f_η with Lipschitz constant

$$L_\eta := \|A\|^2/(\eta\sigma_v), \tag{7}$$

where $\|A\|$ is the operator norm of A. Moreover, the "closeness" of $f_\eta(\cdot)$ to $f(\cdot)$ depends linearly on the smoothing parameter η. More precisely,

$$f_\eta(x) \le f(x) \le f_\eta(x) + \eta D_{v,Y}, \ \forall x \in X, \tag{8}$$

where the "diameter" of Y under the Bregman distance indued by v is defined as

$$D_{v,Y} := \max_{y,z \in Y} \{v(y) - v(z) - \langle \nabla v(z), y - z \rangle\}. \tag{9}$$

In this paper, we assume that (1) has at least one solution, and let x^* denote any solution of (1) and $F^* := F(x^*)$ be the optimal objective function value. We also assume that there exist a first order oracle to compute $f_0(x)$, $f(x)$ and $g_i(x)$, as well as some $\xi_0(x) \in \partial f_0(x)$, $\xi(x) \in \partial f(x)$ and $\zeta_i(x) \in \partial g_i(x)$, for any given $x \in X$. Then our goal is, for any prescribed tolerance $\epsilon > 0$, to find an ϵ-solution x_ϵ to (1) such that

$$F(x_\epsilon) - F^* \le \epsilon \quad \text{and} \quad g_i(x_\epsilon) \le \epsilon, \ \forall\, i = 1, \dots, n_c. \tag{10}$$

The functional constrained problem (1) is considered very challenging especially if g_i's are not simple and/or the projection onto the feasible set $\{x \in X : g_i(x) \le 0, \forall\, i\}$ is difficult to compute [1]. In recent years, we have witnessed a fast development of level-bundle methods for solving (1), which employ historical information and bundle management techniques in a sophisticated manner to achieve very promising efficiency. For a series of important work on level-bundle methods, we refer to [8, 11, 13, 17, 27].

For notation simplicity, we demonstrate the proposed level-bundle methods with only one constraint $g(x) \le 0$, as it is straightforward to rewrite our results for multiple-constraint case: the changes are, for example, from the improvement function $h(x, L) := \max\{f(x) - L, g(x)\}$ with one constraint to $h(x, L) := \max\{f(x) - L, g_1(x), \dots, g_{n_c}(x)\}$ for multiple constraints, etc. Note that ρ_g in the present work is interpreted as $\min_i\{\rho_i\}$, which is only bottlenecked by the least smooth constraint function among g_i's. This is in contrast to existing level-bundle methods designed to work for nonsmooth constraint, where multiple constraints $g_i \le 0$ for all i are reduced to a single constraint $g(x) := \max_i\{g_i(x)\} \le 0$. As a consequence, g can be nonsmooth ($\rho_g = 0$) even if all g_i's are smooth ($\min_i\{\rho_i\} = 1$). This artificial reduction from g_i's to g does not make difference for these methods, but yields in a worse iteration complexity bound than that with g_i's treated separately in our method.

1.2 Related work

In contrast to widely used gradient-descent methods and their numerous variants, level-bundle methods are based on a very different approach to find an ϵ-solution by tightening the gap between the upper and lower bounds of the optimal value of the objective function. The so-called cutting plane model plays an important role in generating those bounds in level-bundle methods. For the convex programming (CP) problem as follows,[1]

$$f_X^* := \min_{x\in X} f(x), \tag{11}$$

with given $x_1, x_2, \ldots, x_k \in X$, the cutting plane model is defined by

$$m_k^f(x) := \max\{\ell_f(x_i, x), 1 \le i \le k\}, \tag{12}$$

where a cutting plane $\ell_f(z, \cdot)$ of f at z is defined by

$$\ell_f(z, x) := f(z) + \langle \xi(z), x - z\rangle \tag{13}$$

for some $\xi(z) \in \partial f(z)$. Therefore (12) bounds the objective function $f(\cdot)$ from below due to the convexity of f. The classical level-bundle method proposed by Lemaréchal, Nemirovskii and Nesterov [17] defined the basic framework of level-bundle methods–given $x_1, x_2, \ldots, x_k$, the classical level-bundle method [17] performs the following three steps in each iteration:

a. Set $\overline{f}_k := \min\{f(x_i), 1 \le i \le k\}$ and compute $\underline{f}_k = \min_{x\in X} m_k^f(x)$ as the upper and lower bounds of f_X^*, respectively.
b. Set the level $l_k = \beta \underline{f}_k + (1-\beta)\overline{f}_k$ for some $\beta \in (0, 1)$.
c. Set $X_k := \{x \in X : m_k^f(x) \le l_k\}$ and determine a new iterate by solving

$$x_{k+1} = \arg\min_{x\in X_k} \|x - x_k\|^2. \tag{14}$$

We can see that the upper bounds $\{\overline{f}_k\}$ and lower bounds $\{\underline{f}_k\}$ on f_X^* are monotonically decreasing and increasing, respectively, and the gaps between them are tightened. If the termination condition is set to $\overline{f}_k - \underline{f}_k \le \epsilon$ where $\overline{f}_k = f(x_k)$ for some x_k, then $0 \le f(x_k) - f_X^* \le \overline{f}_k - \underline{f}_k \le \epsilon$, which means that x_k is an ϵ-solution of (11).

In recent years, there have been increasing research interests in improving iteration complexity of level-bundle type methods for solving smooth CPs inspired by the development of accelerated gradient decent methods. In [16], Nesterov's accelerated

[1] Within Sect. 1.2, we do not separate the objective function as in (1) and refer f to the entire objective function of the optimization.

multi-sequence scheme for smooth CPs [20, 23] and the smoothing technique [22] for non-smooth CPs are employed to improve iteration complexity of level-bundle methods for unconstrained convex optimization problems where the objective functions are (weakly) smooth or in an important class of saddle-point (SP) problems. In [4], the performance of these acceleration methods are further improved by simplified schemes and more efficient subproblem solvers. Moreover, these accelerated algorithms are extended to unconstrained CP problems where X is unbounded. For more details about the developments of level-bundle methods we refer to [7, 16, 17]. Recently, several level-bundle methods with the incorporation of Nesterov's multi-sequence scheme for smooth CPs and a class of saddle point problems using inexact oracle are developed [3]. The accuracy of the approximate solution and the convergence analysis for those algorithms are also studied.

Level-bundle methods are also developed to solve functional constrained convex optimization problem (1) where f and g_i's can be nonsmooth [8, 11, 13, 17, 27]. The idea shared by these methods is to convert the constrained problem (1) to an equivalent, unconstrained problem, for which the classical level-bundle method described above can be applied with necessary modifications. For example, in [13, 27], restricted-memory variants [5, 12, 14] are employed in level-bundle method to solve the following problem which is equivalent to (1):

$$\min_{x\in X} h^*(x), \quad \text{where} \quad h^*(x) := \max\{f(x) - f^*, g(x)\}. \tag{15}$$

We can see that x_ϵ is an ϵ-solution to (15) if and only if $f(x_\epsilon) - f^* \le \epsilon$ and $g(x_\epsilon) \le \epsilon$, i.e., x_ϵ is an ϵ-solution to (1) in the sense of (10). However, the optimal value f^* is unknown in practice and hence one cannot obtain the first-order information of $h^*(x)$ directly. To tackle this issue, a non-decreasing sequence $\{L_k\}$ is generated such that $L_k \uparrow f^*$ and used in place of f^* in (15) in each iteration k [13, 27]. Namely, $h^*(x)$ in (15) is replaced by

$$h(x, L_k) := \max\{f(x) - L_k, g(x)\}, \tag{16}$$

for some $L_k \le f^*$. Note that $h(x, L_k) \ge h^*(x) \ge 0$ for any $x \in X$. In addition, it is easy to see that $h^*(x^*) = 0$ and therefore $h^*(x)$ has a tight lower bound 0. If we define

$$\bar{h}_k := \min\{h(x_i, L_k) : i = 1, \dots, k\}, \tag{17}$$

then $\bar{h}_k$ is the current best estimate of $\min_{x\in X} h(x, L_k)$. Therefore, the goal is to generate $\{x_k\}$ and $\{L_k\}$ such that $\bar{h}_k \downarrow 0$ [13, 27]. To this end, [13] computes

$$L_k := \min\{m_k^f(x) : m_k^g(x) \le 0, x \in X\}, \tag{18}$$

for given $x_1, \dots, x_k$, where $m_k^f(x)$ and $m_k^g(x)$ are the current cutting plane models of $f(\cdot)$ and $g(\cdot)$ defined in (12), respectively. Then a level $l_k := (1 - \beta)\bar{h}_k$ is set, where $\bar{h}_k$ is defined in (17), and the next iterate x_{k+1} is obtained by projecting x_k to the level set $X_k := \{x \in X : m_k^f(x) - L_k \le l_k, m_k^g(x) \le 0\}$. This algorithm achieves the iteration complexity of $O(\epsilon^{-2}\log\epsilon)$. Different from [13], the constrained level-bundle method in [27] generates the sequence $\{L_k\}$ and iterates $\{x_k\}$ jointly without solving (18). Instead, it projects x_k to the level set X_k–if no feasible solution x_{k+1} can be found, then it enlarges the level set by increasing L_k to $L_k + l_k$, and repeats this procedure until a new iterate x_{k+1} is found. In [8, 17], the constrained problem (1) is reformulated to an equivalent min-max problem as follows using the duality theory:

$$\min_{x \in X} \max_{0 \le \alpha \le 1} h(x, \alpha) \quad \text{where} \quad h(x, \alpha) := \alpha(f(x) - f^*) + (1 - \alpha)g(x). \tag{19}$$

Then, at each iteration k, the unknown f^* is replaced by its lower bound L_k in (18), namely, $h(x, \alpha)$ is replaced by

$$h(x, \alpha, L_k) := \alpha(f(x) - L_k) + (1 - \alpha)g(x),$$

and α and x are updated alternately in the just discussed algorithms. For fixed α_k at each iteration k, the classical level-bundle method is applied to $h_k(x, \alpha_k; L_k)$. Similar as the constrained level-bundle methods discussed above, $h(x^*, \alpha_k, L_k)$ is bounded below by 0 and above by $\bar{h}_k := \min\{h(x_i, \alpha_k, L_k) : i = 1, \ldots, k\}$, which is therefore the gap between the upper and lower bounds of $\min_{x \in X} h(x, \alpha_k, L_k)$ Then a level set $X_k := \{x \in X : \alpha_k(m_k^f(x) - L_k) + (1 - \alpha_k)m_k^g(x) \le l_k\}$, where $l_k := (1 - \beta)\bar{h}_k$, is built for $h(x, \alpha_k; L_k)$. With similar idea, the method in [11] further incorporates the bundle aggregation technique and a filter strategy for evaluating candidate points for solving $\min_{x \in X} h(x, \alpha_k; L_k)$. In [6], a bundle method is combined with the target radius method to solve nonsmooth convex optimization where the constraint is on the boundedness of a specific strongly convex function, achieving iteration complexity $\mathcal{O}(\epsilon^{-2})$.

To obtain an ϵ-solution to (1) with nomsmooth $f(\cdot)$ and $g(\cdot)$, the algorithms in [8, 13, 17] and [27] exhibit iteration complexities $\mathcal{O}(\epsilon^{-2}\log(\epsilon))$ and $\mathcal{O}(\epsilon^{-3})$, respectively. Moreover, [8, 9] and [27] further extend the constrained level-bundle methods in [17] and [13] to deal with inexact oracles, respectively.

1.3 Contribution

Our main contribution of this work lies in the development of a unified level-bundle method, called the accelerated constrained level bundle (ACLB), for solving the constrained convex optimization problem (1). By employing Nesterov's accelerated gradient technique, we show that ACLB can maintain the same iteration complexity as the existing level-bundle methods for nonsmooth problems, while significantly improves the complexity if the objective and constraint functions are (weakly) smooth. We also show that, if the objective function has a nonsmooth component which can be written as a max-type function, the iteration complexity for this component can be much lower than that as if we treat it as a generic nonsmooth function. In summary, compared to existing level-bundle methods, the proposed algorithms enjoy orders of lower iteration complexities when the objective and constraint functions are (weakly) smooth. More specifically, ACLB attains the iteration complexity $\mathcal{O}(\epsilon^{-3(1+\rho)/(1+3\rho)} + \|A\|\epsilon^{-2})$, where $\rho = \min\{\rho_i : i = 0, 1, \ldots, n_c\}$ is the least smooth coefficient among the functions f_0 and g_i's in the problem (1). A comparison with existing methods with respect to iteration complexity in two extreme cases is given in Table 1.

Table 1 Iteration complexity of the existing best and our ACLB method on two extreme cases $\rho = 0$ and $\rho = 1$. Note that $\rho = \min\{\rho_i : i = 0, 1, \dots, n_c\}$ for ACLB

Smoothness	Existing	ACLB
$\rho = 0$	$\mathcal{O}(\epsilon^{-2})$ [6], $\mathcal{O}(\epsilon^{-2}\log\epsilon)$ [13] or $\mathcal{O}(\epsilon^{-3})$ [27]	$\mathcal{O}(\epsilon^{-3})$
$\rho = 1$	Same as above	$\mathcal{O}(\epsilon^{-3/2})$

Given the improved iteration complexity of ACLB for (weakly) smooth problems, however, we also point out a drawback of our methods: besides every call of the standard first order oracle, which computes both function values and (sub)gradients of f and g_i's, our methods require an additional function evaluation of f and g_i's. In other words, each iteration of our algorithms calls the zeroth order oracle and the first order oracle, each once. This is in contrast to most existing level-bundle methods where each iteration only calls the first order oracle once. Therefore, the actual per-iteration cost of ACLB is higher (but no more than twice higher) than that of those existing level-bundle methods. For certain problems, such as unit-commitment, Lagrangian relaxation, and two-stage stochastic problems, a subgradient is a byproduct of the function evaluation and hence the zeroth order and first order oracles are at the same cost. Meanwhile, there are also many problems where function evaluations are computationally much less expensive than (sub)gradient evaluations, for which the increase of our per-iteration cost due to this additional zeroth order oracle is very minor. Moreover, the much lowered order of iteration complexity attained by our methods can well compensate such minor increase of per-iteration cost. In addition, our algorithms requires memory space for x^l and x^u besides x, but they are updated not recorded during iterations, which is often a negligible issue in practice.

1.4 Paper organization

The remainder of this paper is organized as follows. In Sect. 2, we present our ACLB algorithm in details. In Sect. 3, we provide a comprehensive analysis of the iteration complexity of ACLB. Numerical results of ACLB are presented in Sect. 4. Section 5 concludes this paper.

2 Accelerated constrained level-bundle method

In this section, we provide a detailed description of our proposed accelerated level-bundle (ACLB) method. To tackle the constraint, ACLB relies on the improvement function h defined by

$$h(x, L) := \max\{F(x) - L, g(x)\}, \tag{20}$$

where L is a lower bound of the unknown optimal value F^*, i.e., $L \le F^*$. Note that there is $h(x, L) \ge 0$ for any $x \in X$. The goal of ACLB is thus to generate a sequence of pairs of iterate and lower bound (x_n, L_n), such that the lower bounds $L_n \uparrow F^*$ and

$h(x_n, L_n) \downarrow 0$ as $n \to \infty$. Therefore, at the heart of ACLB is a "gap reduction" procedure $\mathcal{G}_{\text{ACLB}}$. Let $\beta, \theta \in (0,1)$ be arbitrary and fixed, then given an input triple (x, L, Δ) where $\Delta := h(x, L) \geq 0$ is the *gap*, the gap reduction procedure $\mathcal{G}_{\text{ACLB}}$ can output (x^+, L^+, Δ^+) such that either the new lower bound L^+ improves over L in the sense that $L^+ = L + (1-\beta)\Delta \in (L, F^*)$, or $L^+ = L$ but the new gap Δ^+ is reduced in the sense that $\Delta^+ := h(x^+, L^+) \leq (1 - \beta + \theta\beta)\Delta = q\Delta$ where $q := 1 - \beta + \theta\beta \in (0,1)$. The ACLB gap reduction procedure $\mathcal{G}_{\text{ACLB}}$ is given in Procedure 1.

Procedure 1 ACLB gap reduction procedure $\mathcal{G}_{\text{ACLB}}$: $(x^+, L^+, \Delta^+) = \mathcal{G}_{\text{ACLB}}(x, L, \Delta, \beta, \theta, D_{v,Y})$

1: (Initialization) Denote $F_\eta(x) := f_0(x) + f_\eta(x)$ and define $h_\eta(x) = \max\{F_\eta(x) - L, g(x)\}$, where

$$\eta := \beta\theta\Delta/(2D_{v,Y}). \tag{21}$$

Set $\overline{h}_0 = \Delta$, $l = (1-\beta)\Delta$, $R_0 = X$, $x_0^u = x_0 = x$, $k = 1$.

2: (Update the cutting plane model) Set

$$x_k^l = (1-\alpha_k)x_{k-1}^u + \alpha_k x_{k-1}, \tag{22}$$

$$\underline{R}_k = \{x \in R_{k-1} : \ell_{F_\eta}(x_k^l, x) - L \leq l,\ \ell_g(x_k^l, x) \leq 0\} \tag{23}$$

3: (Update the iterate or lower bound of F^*) Solve x_k from

$$x_k = \arg\min_{z \in \underline{R}_k} \left\{ d(x,z) := \tfrac{1}{2}\|z - x\|^2 \right\}. \tag{24}$$

If no solution, i.e., $\underline{R}_k = \emptyset$, then **terminate** and **output** $x^+ = x_{k-1}^u$, $L^+ = L + l$, $\Delta^+ = h(x^+, L^+)$.

4: (Update the upper bound) Set

$$\tilde{x}_k^u = (1-\alpha_k)x_{k-1}^u + \alpha_k x_k, \tag{25}$$

$$x_k^u = \begin{cases} \tilde{x}_k^u, & \text{if } h_\eta(\tilde{x}_k^u, L) < h_\eta(x_{k-1}^u, L), \\ x_{k-1}^u, & \text{otherwise}, \end{cases} \tag{26}$$

and set $\overline{h}_k = h(x_k^u, L)$. If $\overline{h}_k - l \leq \beta\theta\Delta$, then **terminate** and **output** $x^+ = x_k^u$, $L^+ = L$, $\Delta^+ = \overline{h}_k$.

5: (Bundle management) Define $\overline{R}_k$ and choose R_k satisfying $\underline{R}_k \subseteq R_k \subseteq \overline{R}_k$, where

$$\overline{R}_k := \{z \in X : \langle x_k - x, z - x_k \rangle \geq 0\}. \tag{27}$$

Set $k = k + 1$ and go to Step 2.

The key of the acceleration property of ACLB procedure 1 is due to the Nesterov's acceleration technique. In this case, we need a sequence of combination parameters $\{\alpha_k\} \subset \mathbb{R}$ to satisfy the following properties: there exist $c_1, c_2 > 0$ such that

$$\alpha_1 = 1, \quad 0 < \alpha_k \leq 1, \quad \frac{c_1}{k} \leq \alpha_k \leq \frac{c_2}{k}, \quad \text{and} \quad \frac{1-\alpha_{k+1}}{\alpha_{k+1}^2} \leq \frac{1}{\alpha_k^2}, \quad \forall k \geq 1. \tag{28}$$

The following proposition provides two examples of such $\{\alpha_k\}$.

Proposition 1 *The sequence* $\{\alpha_k\}$ *generated by either way below satisfies* (28) *with* $c_1 = 1$ *and* $c_2 = 2$:

a. $\alpha_k = \frac{2}{k+1}$ *for* $k \geq 1$.
b. $\alpha_k > 0$ *is recursively defined by*

$$\alpha_1 = 1, \quad \alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2, \quad \forall k \geq 1, \tag{29}$$

Proof Part (a) can be verified directly by checking $\alpha_k = \frac{2}{k+1}$ in (28).

For part (b), it is easy to show by induction that $\alpha_k \in (0,1]$ and $\alpha_{k+1} < \alpha_k$ for all $k \geq 1$. Since (29) implies that $\frac{1}{\alpha_{k+1}} - \frac{1}{\alpha_k} = \frac{\alpha_k - \alpha_{k+1}}{\alpha_k \alpha_{k+1}} = \frac{\alpha_k}{\alpha_k + \alpha_{k+1}}$, we can readily show that $1 > \frac{1}{\alpha_{k+1}} - \frac{1}{\alpha_k} \geq \frac{1}{2}$ for all $k \geq 1$. Noting that $\alpha_1 = 1$, we have $\frac{1}{\alpha_k} = 1 + \sum_{i=1}^{k-1}(\frac{1}{\alpha_{i+1}} - \frac{1}{\alpha_i})$, which is bounded between $1 + \frac{k-1}{2} = \frac{k+1}{2}$ and $1 + (k-1) = k$. Therefore $\frac{1}{k} < \alpha_k \leq \frac{2}{k+1} < \frac{2}{k}$. □

We now add a few remarks about $\mathcal{G}_{\text{ACLB}}$ (Procedure 1). Firstly, the linear approximation of $F_\eta(\cdot)$ instead of $F(\cdot)$ are used to define $\underline{R}_k$ if the max-type function f presents in the objective function of (1). However, the definition of $\bar{h}_k$ and the termination condition in Step 4 are still defined on the upper bound on $h(x, L)$. Secondly, the parameter η due to the presence of f is specified as a function of the input Δ and the parameters β, θ, and $D_{v,Y}$ (or any user-chosen value greater than $D_{v,Y}$), and it is fixed within each $\mathcal{G}_{\text{ACLB}}$ and decreases in the input gap Δ. Thirdly, the $\mathcal{G}_{\text{ACLB}}$ either terminates at Step 3 to increase L or at Step 4 to reduce Δ.

Our ACLB gap reduction procedure $\mathcal{G}_{\text{ACLB}}$ (Procedure 1) also employs bundle management (Step 5), also known as bundle compression or restricted memory in the literature, to maintain finite bundle size and hence ensure implementation feasibility in practice. This is realized by the flexible choice of localizer R_k in Step 5: one can discard some linear inequality constraints in $\underline{R}_k$ to obtain R_k, as long as the latter still lies in the half space defined by $\overline{R}_k$ in (27). Note that, in addition to the cost of first order oracles, bundle methods (including ACLB) also require solving a quadratic program (QP) in each iteration which can be costly for problems with high dimensionality (large d_X). However, with small bundle size (e.g., 5-10), one can instead solve the dual problem of QP with very low computational cost [4].

Finally we are ready to present the ACLB method in Algorithm 1.

Algorithm 1 Accelerated constrained level bundle (ACLB) method

1:	Set tolerance $\epsilon > 0$ and $\beta, \theta \in (0,1)$. If f is present in (1), then also give prox-function $v(\cdot)$ in (5), $D_{v,Y}$ in (9) (or any number greater).
2:	Choose an initial $p_0 \in X$, compute $x_0 \in \arg\min_{x \in X}\{\ell_F(p_0, x) : \ell_g(p_0, x) \leq 0\}$. Set $L_0 = \ell_F(p_0, x_0)$, $\Delta_0 = \max\{F(x_0) - L_0, g(x_0)\}$, and set $n = 0$.
3:	If $\Delta_n \leq \epsilon$, **terminate** and **output** ϵ-solution x_n.
4:	Compute $(x_{n+1}, L_{n+1}, \Delta_{n+1}) = \mathcal{G}_{\text{ACLB}}(x_n, L_n, \Delta_n, \beta, \theta, D_{v,Y})$.
5:	Set $n = n + 1$ and go to Step 3.

3 Convergence analysis

In this section, we establish the iteration complexities of the proposed ACLB (Algorithm 1).

Lemma 2 *Suppose that* $\{x_k^l, x_k, \tilde{x}_k^u, x_k^u\}$ *are generated by* $\mathcal{G}_{\text{ACLB}}$ (*Procedure* 1), *and the procedure does not terminate at the Kth iteration. Then the following estimate holds for the input* Δ:

$$h_\eta(x_k^u) - l \le (1-\alpha_k)(h_\eta(x_{k-1}^u) - l) + \frac{\alpha_k^{1+\rho_0} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} + \frac{\alpha_k^{1+\rho_g} L_g}{1+\rho_g}\|x_k - x_{k-1}\|^{1+\rho_g} + \frac{\alpha_k^2 L_\eta}{2}\|x_k - x_{k-1}\|^2, \tag{30}$$

where $l = (1-\beta)\Delta$ *and* L_η *is the Lipschitz constant of* ∇f_η.

Proof For any $k \ge 1$, we have

$$\begin{aligned}
F_\eta(\tilde{x}_k^u) - L &\le \ell_{F_\eta}(x_k^l, \tilde{x}_k^u) - L \\
&\quad + \frac{L_0}{1+\rho_0}\|\tilde{x}_k^u - x_k^l\|^{1+\rho_0} + \frac{L_\eta}{2}\|\tilde{x}_k^u - x_k^l\|^2 \\
&= (1-\alpha_k)\ell_{F_\eta}(x_k^l, x_{k-1}^u) + \alpha_k \ell_{F_\eta}(x_k^l, x_k) - L \\
&\quad + \frac{\alpha_k^{1+\rho} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} + \frac{\alpha_k^2 L_\eta}{2}\|x_k - x_{k-1}\|^2 \\
&\le (1-\alpha_k)(F_\eta(x_{k-1}^u) - L) + \alpha_k(\ell_{F_\eta}(x_k^l, x_k) - L) + \frac{\alpha_k^{1+\rho} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} \\
&\quad + \frac{\alpha_k^2 L_\eta}{2}\|x_k - x_{k-1}\|^2 \\
&\le (1-\alpha_k)(F_\eta(x_{k-1}^u) - L) + \alpha_k l + \frac{\alpha_k^{1+\rho} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} \\
&\quad + \frac{\alpha_k^2 L_\eta}{2}\|x_k - x_{k-1}\|^2,
\end{aligned}$$

where the first inequality is due to (2), the first equality follows from the definitions of x_k^l in (22) and $\tilde{x}_k^u$ in (25), and the linearity of $\ell_{F_\eta}(x_k^l, \cdot)$, the second inequality is due to the convexity of F_η, and the last follows from (23) and (24). Similarly, for $g(\cdot)$, we have

$$
\begin{aligned}
g(\tilde{x}_k^u) &\le \ell_g(x_k^l, \tilde{x}_k^u) + \frac{L_g}{1+\rho_g}\|\tilde{x}_k^u - x_k^l\|^{1+\rho_g} \\
&\le (1-\alpha_k)\ell_g(x_k^l, x_{k-1}^u) + \alpha_k \ell_g(x_k^l, x_k) + \frac{\alpha_k^{1+\rho_g} L_g}{1+\rho_g}\|x_k - x_{k-1}\|^{1+\rho_g} \\
&\le (1-\alpha_k) g(x_{k-1}^u) + \alpha_k l + \frac{\alpha_k^{1+\rho_g} L_g}{1+\rho_g}\|x_k - x_{k-1}\|^{1+\rho_g},
\end{aligned}
$$

where the last inequality is due to $\ell_g(x_k^l, x) \le 0 < l$. In view of (26), we have $h_\eta(x_k^u) \le h_\eta(\tilde{x}_k^u) = \max\{f(\tilde{x}_k^u) - L, g(\tilde{x}_k^u)\}$. Combining the two inequalities above, we get

$$
\begin{aligned}
h_\eta(x_k^u) \le (1-\alpha_k)\max\{F_\eta(x_{k-1}^u) - L, g(x_{k-1}^u)\} + \alpha_k l + \frac{\alpha_k^{1+\rho_0} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} \\
+ \frac{\alpha_k^{1+\rho_g} L_g}{1+\rho_g}\|x_k - x_{k-1}\|^{1+\rho_g} + \frac{\alpha_k^2 L_\eta}{2}\|x_k - x_{k-1}\|^2.
\end{aligned}
$$

Subtracting l on both sides of the above estimate, we obtain (30). □

The following lemma provides several important properties of the bundle management step in $\mathcal{G}_{\mathrm{ACLB}}$ (Procedure 1).

Lemma 3 *Let* (x, L, Δ) *be the input of* $\mathcal{G}_{\mathrm{ACLB}}$ (*Procedure* 1), *and denote* $\mathcal{E}_l := \{\bar{x} \in X : F(\bar{x}) - L \le l, g(\bar{x}) \le 0\}$ *where* $l = (1-\beta)\Delta$, *then the following statements hold*:

(a) $\underline{R}_k \subseteq \overline{R}_k$ *for all* $k \ge 1$.
(b) *There is* $\mathcal{E}_l \subseteq \underline{R}_k \subseteq R_k \subseteq \overline{R}_k$ *for all* $k \ge 1$. *If* $\mathcal{E}_l \ne \emptyset$, *then* (24) *has a unique solution.*
(c) *If* $\underline{R}_k = \emptyset$, *then* $L + l < f^*$.
(d) *If terminated in Step 4, then* $\Delta^+ \le q\Delta$ *where* $q := 1 - \beta + \theta\beta$.

Proof

(a) If x in (24) satisfies $x \in \underline{R}_k$, then due to the fact that x_k is the projection of x onto $\underline{R}_k$ in (24), we know $x_k = x$, and $\langle x_k - x, z - x_k\rangle = 0$ for all $z \in X$. Therefore $\overline{R}_k = X$ and $\underline{R}_k \subseteq \overline{R}_k$. If $x \notin \underline{R}_k$, then due to the optimality condition of x_k in (24), we have $\|z - x\|^2 \ge \|z - x_k\|^2 + \|x_k - x\|^2$ for all $z \in \underline{R}_k$, from which we obtain $\langle x_k - x, z - x_k\rangle \ge 0$, i.e., $z \in \overline{R}_k$. Hence $\underline{R}_k \subseteq \overline{R}_k$.
(b) We prove the result by induction. Since $R_0 = X$, there is $\mathcal{E}_l \subseteq R_0$. Assume that $\mathcal{E}_l \subseteq R_{k-1}$ holds for some $k \ge 1$. Note that for any $x \in \mathcal{E}_l$, we have $\ell_{F_\eta}(x_k^l, x) - L \le F_\eta(x) - L \le F(x) - L \le l$ and $\ell_g(x_k^l, x) \le g(x) \le 0$ in $\mathcal{G}_{\mathrm{ACLB}}$ due

to the convexity of f, f_η, and g. By the definition of $\underline{R}_k$ in (23), and the induction assumption $\mathcal{E}_l \subseteq R_{k-1}$, we have $\mathcal{E}_l \subseteq \underline{R}_k$. Due to $\underline{R}_k \subseteq \overline{R}_k$ in Part (a) and the choice of R_k such that $\underline{R}_k \subseteq R_k \subseteq \overline{R}_k$ for any $k \geq 1$, we obtain $\mathcal{E}_l \subseteq R_k$. Therefore we have $\mathcal{E}_l \subseteq \underline{R}_k \subseteq R_k \subseteq \overline{R}_k$ by induction. Moreover, $d(x)$ is strongly convex and $\underline{R}_k$ is nonempty, hence (24) has a unique solution.

(c) If $L + l \geq F^*$, then for every solution x^* to (1), there is $F(x^*) - L = F^* - L \leq l$ and $g(x^*) \leq 0$, and hence $x^* \in \mathcal{E}_l$, which contradicts to $\underline{R}_k = \emptyset$. Therefore $L + l < F^*$.

(d) The inequality $\Delta^+ \leq q\Delta$ follows immediately due to the definition of $l = (1-\beta)\Delta$ and the termination condition $\overline{h}_k - l \leq \beta\theta\Delta$ in Step 4.

□

Now we have the following bound for the iterates $\{x_k\}$ within each $\mathcal{G}_{\mathrm{ACLB}}$.

Lemma 4 *Let $\{x_k\}$ be the iterates generated by $\mathcal{G}_{\mathrm{ACLB}}$ before termination at some iteration K, then*

$$\sum_{k=1}^{K} \|x_k - x_{k-1}\|^2 \leq D_X^2, \tag{31}$$

where the diameter D_X of the compact set X is defined by

$$D_X := \max_{x,y\in X} \|x - y\|. \tag{32}$$

Proof For all $k > 1$, there is $x_k \in \underline{R}_k \subseteq R_{k-1} \subseteq \overline{R}_{k-1}$, where the first inclusion is due to the definition of x_k in (24), the second due the definition of $\underline{R}_k$ in (23), and the last due to the selection $R_k \subseteq \overline{R}_k$ in Step 5 of $\mathcal{G}_{\mathrm{ACLB}}$ for all k. Furthermore, for any input x in $\mathcal{G}_{\mathrm{ACLB}}$, we know $d(x,z) = (1/2)\cdot\|z - x\|^2$ is strongly convex in z with modulus 1. Therefore,

$$d(x,x_k) \geq d(x,x_{k-1}) + \langle x_{k-1} - x, x_k - x_{k-1}\rangle + \frac{1}{2}\|x_k - x_{k-1}\|^2 \geq d(x,x_{k-1}) + \frac{1}{2}\|x_k - x_{k-1}\|^2,$$

where the last inequality follows from $x_k \in \overline{R}_{k-1}$ as we showed earlier and the definition of $\overline{R}_k$ in (27). By taking the sum of both sides over $k = 1, 2, \ldots, K$, we obtain

$$\frac{1}{2}\sum_{k=1}^{K} \|x_k - x_{k-1}\|^2 \leq d(x,x_K) - d(x,x_0) \leq d(x,x_K) = \frac{1}{2}\|x_K - x\|^2 \leq \frac{1}{2}D_X^2,$$

which implies the bound (31). □

Proposition 5 *If $\{\alpha_k\}$ in $\mathcal{G}_{\mathrm{ACLB}}$ is chosen such that (28) holds, then the number of iterations performed within each $\mathcal{G}_{\mathrm{ACLB}}$ (Procedure 1) with input Δ does not exceed*

$$N_{\mathrm{ACLB}}^{\mathrm{inner}}(\Delta) := C\left((\frac{L_0}{\Delta})^{2/(1+3\rho_0)} + (\frac{L_g}{\Delta})^{2/(1+3\rho_g)} + (\frac{\|A\|}{\Delta}\sqrt{\frac{D_{v,Y}}{\sigma_v}}) \right) = \mathcal{O}\left(\frac{L_0}{\Delta})^{2/(1+3\rho_0)}) + \mathcal{O}((\frac{L_g}{\Delta})^{2/(1+3\rho_g)}) + \mathcal{O}((\frac{\|A\|}{\Delta}) \right), \tag{33}$$

where $C > 0$ is the constant dependent on ρ_0, ρ_g, D_X, θ and β only, and D_X and $D_{v,Y}$ are defined in (32)and (9) respectively.

Proof Suppose that $\mathcal{G}_{\mathrm{ACLB}}$ does not terminate at the Kth iteration for some $K > 0$. Dividing both sides of (30) by α_k^2, we obtain that

$$\frac{h_\eta(x_k^u) - l}{\alpha_k^2} \le \frac{(1-\alpha_k)(h_\eta(x_{k-1}^u) - l)}{\alpha_k^2} + \frac{\alpha_k^{\rho_0 - 1} L_0}{1+\rho_0}\|x_k - x_{k-1}\|^{1+\rho_0} + \frac{\alpha_k^{\rho_g - 1} L_g}{1+\rho_g}\|x_k - x_{k-1}\|^{1+\rho_g} + \frac{L_\eta}{2}\|x_k - x_{k-1}\|^2. \tag{34}$$

Noting that α_k satisfies (28), and taking sum of k over $1, \ldots, K$, on both sides of (34), we have

$$\frac{h_\eta(x_K^u) - l}{\alpha_K^2} \le \frac{L_0}{1+\rho_0}\sum_{k=1}^K \alpha_k^{\rho_0-1}\|x_k - x_{k-1}\|^{1+\rho_0} + \frac{L_g}{1+\rho_g}\sum_{k=1}^K \alpha_k^{\rho_g-1}\|x_k - x_{k-1}\|^{1+\rho_g} + \frac{L_\eta}{2}\sum_{k=1}^K \|x_k - x_{k-1}\|^2. \tag{35}$$

By the Hölder's inequality, (31) from Lemma 4, and $\alpha_K > c_1/K$ for some $c_1 > 0$, we obtain

$$\sum_{k=1}^K \alpha_k^{\rho-1}\|x_k - x_{k-1}\|^{1+\rho} \le \left(\sum_{k=1}^K \alpha_k^{-2}\right)^{\frac{1-\rho}{2}} \left(\sum_{k=1}^K \|x_k - x_{k-1}\|^2\right)^{\frac{1+\rho}{2}} \le C(K^{\frac{3-3\rho}{2}})D_X^{1+\rho}, \tag{36}$$

where $C > 0$ is a constant depending only on ρ and c_1. By (36) with ρ substituted by ρ_0 and ρ_g, and $c_1/K < \alpha_K \le c_2/K$, we deduce from (35) that

$$h_\eta(x_K^u) - l \le CL_0K^{-\frac{1+3\rho_0}{2}}D_X^{1+\rho_0} + L_gK^{-\frac{1+3\rho_g}{2}}D_X^{1+\rho_g} + \frac{L_\eta}{2}K^{-2}D_X^2 \tag{37}$$

Then by (8) and (21), we can obtain

$$\bar{h}_K - l = h(x_K^u) - l \le h_\eta(x_K^u) - l + \eta D_{v,Y} \le CL_0K^{-\frac{1+3\rho_0}{2}}D_X^{1+\rho_0} + L_gK^{-\frac{1+3\rho_g}{2}}D_X^{1+\rho_g} + \frac{L_\eta}{2}K^{-2}D_X^2 + \frac{\theta\beta\Delta}{2},$$

where we used (35) and $\eta = \beta\theta\Delta/(2D_{v,Y})$ in (21). In view of the termination condition in Step 4 of $\mathcal{G}_{\text{ACLB}}$ Procedure 1, we have $\bar{h}_k - l > \theta\beta\Delta$, therefore,

$$\frac{\theta\beta\Delta}{2} \leq 3C \max L_0 K^{-\frac{1+3\rho_0}{2}} D_X^{1+\rho_0}, L_g K^{-\frac{1+3\rho_g}{2}} D_X^{1+\rho_g}, \frac{L_\eta}{2} K^{-2} D_X^2. \tag{38}$$

If the first term in the max in (38) is the largest among the three, then from $\frac{\theta\beta\Delta}{2} \leq 3CL_0 K^{-\frac{1+3\rho_0}{2}} D_X^{1+\rho_0}$, we have $N_{\text{ACLB}}^{\text{inner}}(\Delta) = C(\frac{L_0}{\Delta})^{2/(1+3\rho_0)} = \mathcal{O}((\frac{L_0}{\Delta})^{2/(1+3\rho_0)})$. Similar argument holds if the second term in the max in (38) is the largest. If the third term in the max in (38) is the largest, then by the fact that $L_\eta = \|A\|^2/(\eta\sigma_v) = 2\|A\|^2 D_{v,Y}/(\theta\beta\sigma_v\Delta)$ in (7), we have $N_{\text{ACLB}}^{\text{inner}}(\Delta) = C\frac{\|A\|}{\Delta}\sqrt{C\frac{D_{v,Y}}{\sigma_v}} = \mathcal{O}(\frac{\|A\|}{\Delta})$. Therefore (33) holds. □

Finally, we are ready to establish the iteration complexity of ACLB (Algorithm 1). For ease of presentation, we call a gap reduction procedure $\mathcal{G}_{\text{ACLB}}$ (Procedure 1) *critical* if $\mathcal{G}_{\text{ACLB}}$ terminates at Step 4, i.e., $\bar{h}_k - l \leq \beta\theta\Delta$ in $\mathcal{G}_{\text{ACLB}}$; otherwise it is called *non-critical*, i.e., it terminates at Step 3 and the level set $\underline{R}_k = \emptyset$ in $\mathcal{G}_{\text{ACLB}}$.

Theorem 6 *For any given $\epsilon > 0$, if $\{\alpha_k\}$ in every $\mathcal{G}_{\text{ACLB}}$ (Procedure 1) satisfies (28), then the following statements hold for ACLB (Algorithm 1) to compute an ϵ-solution to problem (1):*

(a) *The total number of calls to $\mathcal{G}_{\text{ACLB}}$ in ACLB (Algorithm 1) does not exceed*

$$N_{\text{ACLB}}^{\text{outer}}(\epsilon) := 1 + \frac{F^* - L_0}{(1-\beta)\epsilon} + \log_{\frac{1}{q}} \frac{\hat{V}_X}{\epsilon} = \mathcal{O}\frac{1}{\epsilon}, \tag{39}$$

where the constant $\hat{V}_X$ independent of ϵ is defined by

$$\hat{V}_X := \max 2\|A\|\sqrt{2D_{v,Y}/\sigma_v}, \frac{L_g}{1+\rho_g} D_X^{1+\rho_g}, \frac{L_0}{1+\rho_0} D_X^{1+\rho_0}. \tag{40}$$

(b) *The total number calls to the first order oracle in ACLB (Algorithm 1) does not exceed*

$$N_{\text{ACLB}}^{\text{total}}(\epsilon) := N_{\text{ACLB}}^{\text{inner}}(\epsilon) \cdot N_{\text{ACLB}}^{\text{outer}}(\epsilon) \leq \mathcal{O}(L_0^{\frac{2}{1+3\rho_0}} \epsilon^{-\frac{3(1+\rho_0)}{1+3\rho_0}}) + \mathcal{O}(L_g^{\frac{2}{1+3\rho_g}} \epsilon^{-\frac{3(1+\rho_g)}{1+3\rho_g}}) + \mathcal{O}(\|A\|\epsilon^{-2}). \tag{41}$$

Proof

(a) We can partition the set of iteration counters in the ACLB Algorithm 1 into $\{i_1, \ldots, i_{\bar{m}}\}$ for non-critical calls of $\mathcal{G}_{\text{ACLB}}$ and $\{j_1, \ldots, j_{\bar{n}}\}$ for critical calls of $\mathcal{G}_{\text{ACLB}}$, where $\bar{N} = \bar{n} + \bar{m}$ is the total number of calls of $\mathcal{G}_{\text{ACLB}}$ in ACLB (Algorithm 1).

Note that $\mathcal{G}_{\text{ACLB}}$ with input Δ will output $\Delta^+ \leq \Delta$. In addition, if $\mathcal{G}_{\text{ACLB}}$ with input Δ is critical, we have $\Delta^+ \leq q\Delta$ for $q = 1 - \beta + \theta\beta \in (0, 1)$. Therefore, one can easily see that the number of critical $\mathcal{G}_{\text{ACLB}}$'s in Algorithm 1 is finite: we have $\Delta_{j_{m+1}} \le q\Delta_{j_m}$ for all $m = 1, \dots, \bar{m}-1$, and $\Delta_{j_{\bar{m}}} \le \epsilon < \Delta_{j_{\bar{m}-1}}$ due to the termination condition in Step 3 of ACLB (Algorithm 1). Therefore, we have $\epsilon < \Delta_{j_{\bar{m}-1}} \le q^{\bar{m}-2}\Delta_{j_1} \le q^{\bar{m}-2}\Delta_0$, which implies that

$$\bar{m} < \log_{\frac{1}{q}} \frac{\Delta_0}{\epsilon} =: N^{\text{outer-c}}_{\text{ACLB}}(\epsilon) = \mathcal{O}(\log \epsilon). \tag{42}$$

Now we only need to show that $\Delta_0 \le \hat{V}_X$ to finalize the role of (42) in (39), where $\Delta_0 = \max\{F(x_0) - L_0, g(x_0)\}$. From (2) and (3), we have

$$\begin{aligned} F(x_0) - L_0 = F(x_0) - \ell_F(p_0, x_0) &\le \frac{L_0}{1+\rho_0}\|x_0 - p_0\|^{1+\rho_0} + f(x_0) - \ell_f(p_0, x_0) \\ &\le \frac{L_0}{1+\rho_0} D_X^{1+\rho_0} + f(x_0) - \ell_f(p_0, x_0), \end{aligned} \tag{43}$$

$$g(x_0) \le \ell_g(p_0, x_0) + \frac{L_g}{1+\rho_g}\|x_0 - p_0\|^{1+\rho_g} \le \frac{L_g}{1+\rho_g} D_X^{1+\rho_g}, \tag{44}$$

where we used the fact $L_{i_1} = L_0 = \ell_f(p_0, x_0)$ to obtain the equality in (43), and $\ell_g(p_0, x_0) \le 0$ in (44) due to the way we compute x_0 in Step 2 of ACLB (Algorithm 1). Moreover, we have from Lemma 8 in [16] that

$$f(x_0) - \ell_f(p_0, x_0) \le 2\|A\|\sqrt{2D_{v,Y}/\sigma_v}. \tag{45}$$

Combining (43), (44) and (45), we obtain $\Delta_0 \le \hat{V}_X$, where $\hat{V}_X$ is defined in (40). Therefore, we have

$$N^{\text{outer-c}}_{\text{ACLB}}(\epsilon) = \bar{m} < \log_{\frac{1}{q}} \frac{\hat{V}_X}{\epsilon} = \mathcal{O}(\log \epsilon). \tag{46}$$

For each non-critical $\mathcal{G}_{\text{ACLB}}$ with input L, the output L^+ satisfies $L^+ - L = l = (1-\beta)\Delta$. Since input $\Delta_n > \epsilon$ for all n before ACLB Algorithm terminates, we know that $L^+ - L \ge (1-\beta)\epsilon$. Therefore, the number of non-critical $\mathcal{G}_{\text{ACLB}}$'s in ACLB (Algorithm 1) is bounded by

$$N^{\text{outer-nc}}_{\text{ACLB}}(\epsilon) := \frac{F^* - L_0}{(1-\beta)\epsilon} + 1 = \mathcal{O}\frac{1}{\epsilon}. \tag{47}$$

Combining (46) and (47) we obtain that the total number of calls of $\mathcal{G}_{\text{ACLB}}$ in ACLB Algorithm is bounded by $N^{\text{outer}}_{\text{ACLB}}(\epsilon) := N^{\text{outer-c}}_{\text{ACLB}}(\epsilon) + N^{\text{outer-nc}}_{\text{ACLB}}(\epsilon)$, as given in (39).

(b) Now we have known that the number of calls to $\mathcal{G}_{\text{ACLB}}$ in ACLB (Algorithm 1) is bounded above by $N^{\text{outer}}_{\text{ACLB}}(\epsilon)$ in (39). On the other hand, the number of calls to the first order oracle in each $\mathcal{G}_{\text{ACLB}}$ (Procedure 1) in the nth iteration of ACLB (Algorithm 1) is bounded by $N^{\text{inner}}_{\text{ACLB}}(\Delta_n)$ given in (33). Hence, the total number of iterations (calls to the first order oracles of f_0, f and g) is bounded above by

$$\sum_{n=1}^{N} N_{\mathrm{ACLB}}^{\mathrm{inner}}(\Delta_n) \leq N_{\mathrm{ACLB}}^{\mathrm{outer}}(\epsilon) \cdot N_{\mathrm{ACLB}}^{\mathrm{inner}}(\epsilon) = N_{\mathrm{ACLB}}^{\mathrm{total}}(\epsilon), \tag{48}$$

where we used the facts that $\Delta_n > \epsilon$ for all n and $N_{\mathrm{ACLB}}^{\mathrm{inner}}(\Delta)$ is non-decreasing in Δ in the first inequality. Applying (33) and (39) to (48) yields the bound in (41).

□

This theorem provides the iteration complexity bound for each function in the objective functional and constraint. Hence we can have iteration complexity bound for the (1) in various cases.

4 Numerical experiment

In this section, we conduct a series of numerical experiments on synthesized constrained optimization problems. In the first part of our experiments, we evaluate the performance of the proposed ACLB algorithm on a smooth constrained optimization problem, i.e., both of the objective function and constraint function are smooth. We demonstrate the improved convergence rate of ACLB by comparing to a state-of-the-art constrained level-bundle (CLB) method [27] for such smooth constrained problems. In the second part of our experiments, we compare the proposed ACLB on nonsmooth but structured constrained problems, where we implement our method as if the problem is a generic nonsmooth problem (labeled as ACLB) and utilizing the smoothing technique (labeled as ACLB-S). More specifically, the objective function in the latter problem is nonsmooth but can be written as a max-type function using Fenchel duality. We construct a special type of problems under this case, and show that ACLB-S may outperform ACLB by making use of the max-type structure of objective function.

The performance of all comparison algorithms is evaluated using the progresses of improvement function h_k and estimated lower bound L_k where k is the iteration counter (number of calls of the first-order oracle) averaged over 10 random instances. The improvement function h_k should monotonically decrease to 0, and the estimated lower bound L_k should increase to f^* although it is often unknown. Therefore, the algorithm with faster decay (increase) of h_k (L_k respectively) is considered more efficient. All the experiments are implemented and tested in MATLAB R2018a on a Windows desktop with 3.70GHz CPU and 16GB of memory.

4.1 Smooth constrained optimization

We consider the following constrained optimization problem:

$$\min_{x\in X} \frac{1}{2}\|Ax-b\|^2, \quad \text{subject to } \frac{1}{2}\|Cx-d\|^2 \leq e, \tag{49}$$

where $x \in X \triangleq \{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$, A and C are given matrices, b, d are given vectors of compatible dimensions, and $e > 0$ is a given error tolerance. In our experiments, we set both A and C as m-by-n Gaussian random matrices, where

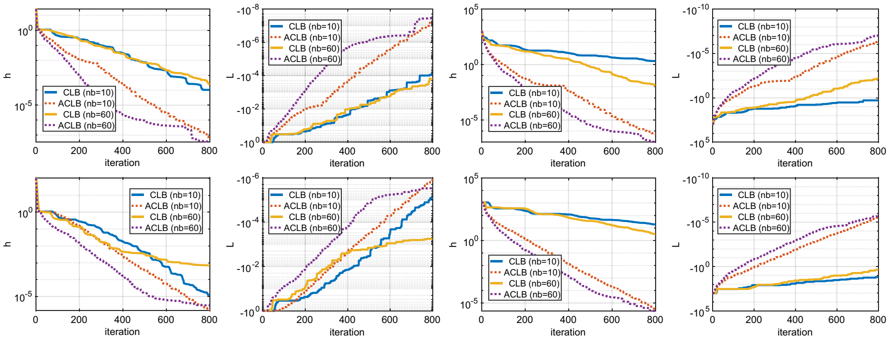

Fig. 1 Average of improvement function h and the corresponding lower bound L over 10 instances versus iteration with initial $L_0 = -1$ (left two columns) and initial $L_0 = -1000$ (right two columns) for the smooth constrained problem (49) with problem sizes $n = 100$ (top row) and $n = 500$ (bottom row)

n is the dimension of the unknown x, and $b, d \in \mathbb{R}^m$, for two different pairs of (m, n) as (50, 100) and (200, 500). More specifically, for each pair of (m, n), we first generate a reference $\bar{x} \in \mathbb{R}^n$ using MATLAB `randn` function and normalize it by $\bar{x} \leftarrow \bar{x}/\|\bar{x}\|$, and then generate two random matrices $A, C \in \mathbb{R}^{m \times n}$, and set $b = Ax$, $d = Cx$, and $e = 10$. Namely, $F(x) = f_0(x) = \frac{1}{2}\|Ax - b\|^2$ and one constraint $g(x) = \frac{1}{2}\|Cx - d\|^2 - e$ in (1). In this case, we know $\bar{x}$ is an optimal solution, and the optimal objective function value is $f^* = 0$. Therefore, a convergent level bundle method applied to (49) should generate $\{x_k\}$ such that $h_k \downarrow 0$ and $L_k \uparrow F^* = 0$ as $k \to \infty$. For comparison, we apply CLB (Algorithm 1 in [27]) and the proposed ACLB algorithm to solve (49). We set $\gamma = 0.6$ for CLB and $\beta = \theta = 0.6$ for ACLB, and tried the initial lower bound of F^* as $L_0 = -1$ and -1000 (denoted by f^0_{low} in CLB [27]), and total bundle size (nb) for both objective and constraint functions to 10 and 60 (i.e., 5 and 30 for each of f and g). The initial x_0 is set to 0, and $\epsilon = 10^{-6}$ in the termination condition (same for all experiments in this section). We also set the same maximum number 800 of calls to the first order oracle in all methods. We use the MATLAB builtin QP solver `quadprog` to solve the subproblems in both algorithms. For $m = 50$ and $n = 100$, we generate 10 random instances, and show the average of h_k and L_k versus iteration number with initial lower bound $L_0 = -1$ (left two plots) in and $L_0 = -1000$ (right two plots) in the top row of Fig. 1. Due to the data generation above, the true $F^* = 0$ in all cases. The results of the same experiment with problem size $m = 200$ and $n = 500$ are shown in the bottom row of Fig. 1. In Table 2, we also show the mean and standard deviation (in parentheses) of L_k, h_k, and CPU time (in seconds) after $k = 800$ iterations over the 10 random instances in the following order from top to bottom: $L_0 = -1$ (first table) and $L_0 = -1000$ (second table) with problem size $n = 100$, and $L_0 = -1$ (third table) and $L_0 = -1000$ (fourth table) with problem size $n = 500$. As we can see, ACLB can significantly improve the convergence for the smooth constrained problem (49).

Table 2 Comparison of CLB and ACLB on the smooth constrained problem (49) with different bundle size (nb)

	L	h	Time
$L_0 = -1,\ n = 100$			
CLB (nb = 10)	−8.25e−5 (2.20e−4)	1.05e−4 (2.31e−4)	2.01e+0 (5.25e−2)
ACLB (nb = 10)	−5.78e−8 (8.18e−8)	7.48e−8 (1.01e−7)	4.23e+0 (2.53e−2)
CLB (nb = 60)	−1.53e−4 (1.53e−4)	2.38e−4 (2.29e−4)	5.14e+0 (3.82e−1)
ACLB (nb = 60)	−3.34e−8 (3.60e−8)	3.57e−8 (3.80e−8)	7.80e+0 (1.90e−1)
$L_0 = -1000,\ n = 100$			
CLB (nb = 10)	−1.93e+0 (6.71e−1)	2.07e+0 (7.31e−1)	1.79e+0 (3.84e−2)
ACLB (nb = 10)	−4.19e−7 (8.21e−7)	5.18e−7 (9.93e−7)	4.39e+0 (1.94e−2)
CLB (nb = 60)	−6.53e−3 (5.39e−3)	1.32e−2 (1.18e−2)	5.30e+0 (1.98e−1)
ACLB (nb = 60)	−9.17e−8 (1.26e−7)	9.95e−8 (1.30e−7)	8.12e+0 (1.82e−1)
$L_0 = -1,\ n = 500$			
CLB (nb = 10)	−7.70e−6 (4.73e−6)	1.22e−5 (8.82e−6)	2.58e+1 (1.35e+0)
ACLB (nb = 10)	−1.08e−6 (9.66e−7)	1.44e−6 (1.48e−6)	2.19e+1 (3.79e−1)
CLB (nb = 60)	−5.59e−4 (1.13e−3)	6.74e−4 (1.24e−3)	4.16e+1 (2.80e+0)
ACLB (nb = 60)	−2.93e−6 (3.89e−6)	2.98e−6 (3.92e−6)	4.19e+1 (2.17e+0)
$L_0 = -1000,\ n = 500$			
CLB (nb = 10)	-1.50e+1 (3.30e+0)	1.92e+1 (9.84e+0)	2.16e+1 (1.16e+0)
ACLB (nb = 10)	−2.94e−6 (2.64e−6)	3.36e−6 (2.94e−6)	2.35e+1 (3.32e−1)
CLB (nb = 60)	−2.30e+0 (1.54e+0)	3.23e+0 (1.80e+0)	4.35e+1 (2.99e+0)
ACLB (nb = 60)	−2.16e−6 (3.18e-6)	2.37e-6 (3.44e-6)	4.54e+1 (1.40e+0)

The values of lower bound L, improvement function h, and the CPU time (in seconds) after 800 iterations using initial $L_0 = -1$ (first table) and $L_0 = -1000$ (second table) with problem size $n = 100$, and initial $L_0 = -1$ (third table) and $L_0 = -1000$ (last table) with problem size $n = 500$

4.2 Structured nonsmooth constrained optimization

We proposed to leverage the special structure of certain nonsmooth constrained problem with improved convergence rate in ACLB, which we call ACLB-S. To demonstrate the improvement gained by ACLB-S, we consider the following nonsmooth constrained problem:

$$\min_{x\in X}\{\|Ax-b\|_1 = \max_{\|y\|_\infty\le 1}\langle Ax-b, y\rangle\}, \quad \text{subject to } \frac{1}{2}\|Cx-d\|^2 \le e, \tag{50}$$

where again $X \triangleq \{x \in \mathbb{R}^n : \|x\|_\infty \le 1\}$, A and C are given matrices, b, d are given vectors of compatible dimensions, and $e > 0$ is a given error tolerance. Therefore, the objective function is $F(x) = f(x) = \max_{y\in Y}\langle Ax, y\rangle - \chi(y)$ where $Y = \{y \in \mathbb{R}^m : \|y\|_\infty \le 1\}$ and $\chi(y) = \langle b, y\rangle$. We apply both ACLB (as if the objective function is a generic nonsmooth function $F(x) = f_0(x) = \|Ax - b\|_1$) and ACLB-S to the problem (50). For this test, both A and C are 2062×2062 matrices, where A is from the worst-case QP instance for first-order methods generated by Nemirovski (see the construction scheme in [19, 21]) and C is a randomly generated using

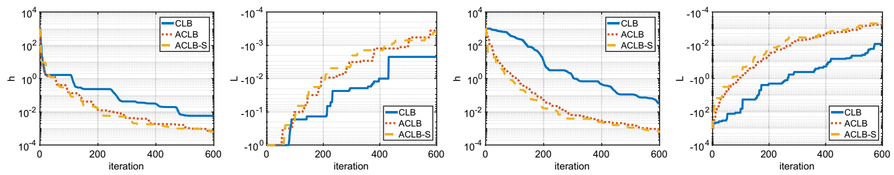

Fig. 2 Average of improvement function h and the corresponding lower bound L over 10 instances for the bundle size 10 versus iteration with initial $L_0 = -1$ (left two plots) and initial $L_0 = -1000$ (right two plots) for the structured nonsmooth constrained problem (50) of size $n = 2062$

Table 3 Comparison of CLB, ACLB, and ACLB-S on the structured nonsmooth constrained problem (50) of size $n = 2062$

	L	h	Time
$L_0 = -1,\ n = 2062$			
CLB	−2.22e−3 (4.31e−4)	5.81e−3 (7.21e−4)	3.97e+2 (2.55e+1)
ACLB	−3.60e−4 (4.93e−4)	7.33e−4 (5.04e−4)	3.18e+2 (8.50e+0)
ACLB-S	−4.02e−4 (3.01e−4)	6.48e−4 (2.84e−4)	3.16e+2 (3.03e+0)
$L_0 = -1000,\ n = 2062$			
CLB	−8.18e−3 (4.29e−3)	3.10e−2 (2.79e−2)	4.25e+2 (2.76e+1)
ACLB	−5.72e−4 (3.12e−4)	9.33e−4 (3.39e−4)	3.43e+2 (1.74e+1)
ACLB-S	−2.71e−4 (1.66e−4)	5.52e−4 (1.53e−4)	3.19e+2 (9.40e+0)

The values of lower bound L, improvement function h, and the CPU time (in seconds) after 600 iterations using initial $L_0 = -1$ (top) and $L_0 = -1000$ (bottom). Bundle size is 10 for all methods

MATLAB builtin function `randn`. Then we set $b = Ax, d = Cx, e = 10^{-3}$, and the initial lower bound of f^* to $L_0 = -1$ and $L_0 = -1000$ for testing. For $L_0 = -1$, we set $\theta = 0.8$ for both ACLB and ACLB-S, and $\beta = 0.5$ for ACLB and 0.4 for ACLB-S. For $L_0 = -1000$, we set $\theta = \beta = 0.6$ for ACLB, and $\theta = 0.7, \beta = 0.4$ for ACLB-S. The total bundle size for both objective and constraint functions to 10. For ACLB-S, we use a sufficiently large estimate 1000 for $D_{v,Y}$ and compute the corresponding η (same below). We also applied CLB Algorithm 1 in [27] with $\gamma = 0.6$ to this problem with the same bundle management setting. These parameter settings seem to yield optimal performance of the comparison algorithms among those we tested. We also set the same maximum number 600 of calls to the first order oracle in all methods. We use the MATLAB builtin QP solver `quadprog` to solve the subproblems in all three algorithms. We again generate 10 random instances. The average of h_k and L_k versus iteration number with initial $L_0 = -1$ (left two plots) and $L_0 = -1000$ (right two plots) are given in Fig. 2. In Table 3, we also show the mean and standard deviation (in parentheses) of L_k, h_k, and CPU time (in seconds) after $k = 600$ iterations with initial $L_0 = -1$ (top table) and $L_0 = -1000$ (bottom table). From these results, we can see ACLB and ACLB-S have similar efficiency for this problem, and both outperform CLB significantly.

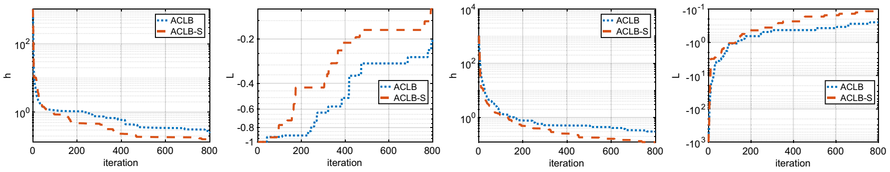

Fig. 3 Average of improvement function h and the corresponding lower bound L over 10 instances for the bundle size 10 versus iteration with initial $L_0 = -1$ (left two plots) and initial $L_0 = -1000$ (right two plots) for the structured nonsmooth constrained problem (50) of size $n = 4124$

Table 4 Comparison of ACLB and ACLB-S on the structured nonsmooth constrained problem (50) of size $n = 4124$

	L	h	Time
$L_0 = -1,\ n = 4124$			
ACLB	−2.00e−1 (1.33e−1)	2.45e−1 (1.36e−1)	1.05e+2 (2.98e+0)
ACLB-S	−1.25e−1 (5.61e−2)	1.35e−1 (5.58e−2)	9.85e+1 (2.75e+0)
$L_0 = -1000,\ n = 4124$			
ACLB	−2.49e−1 (1.68e−1)	3.00e−1 (1.79e−1)	1.05e+2 (4.85e+0)
ACLB-S	−1.15e-1 (3.12e−2)	1.23e−1 (3.13e−2)	9.92e+1 (3.21e+0)

The values of lower bound L, improvement function h, and the CPU time (in seconds) after 800 iterations using initial $L_0 = -1$ (top) and $L_0 = -1000$ (bottom). Bundle size is 10 for all methods

We further consider the problem (50) of where A and C are both 2062×4124, and $X = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$. In this case, the MATLAB builtin QP solver becomes much slower as the dimension of x is $n = 4124$. However, at the small bundle size 10, we can still solve the dual problem efficiently as in [4] as the set X is a ball in $\mathbb{R}^n$. By employing this solver, we apply ACLB and ACLB-S to problem (50) with initial L_0 as -1 and -1000. In both cases, we set $\theta = \beta = 0.5$ for both ACLB and ACLB-S, apply them to 10 randomly generated cases of (50), and compare their performance. We also set the same maximum number 800 of calls to the first order oracle in these methods. The average of h_k and L_k versus iteration number with initial $L_0 = -1$ (left two plots) and $L_0 = -1000$ (right two plots) are given in Fig. 3. In Table 4, we also show the mean and standard deviation (in parentheses) of L_k, h_k, and CPU time (in seconds) after $k = 800$ iterations with initial $L_0 = -1$ (top table) and $L_0 = -1000$ (bottom table). From these results with larger problem size, we can see ACLB-S outperforms ACLB by making use of the max-type structure of the objective function for improved theoretical and practical convergence rate.

5 Concluding remarks

In this paper, we presented an accelerated level bound method, called ACLB, to uniformly solve smooth, weakly smooth and nonsmooth convex constrained optimization problems. We provided convergence analysis of the proposed method. The iteration complexity bound of ACLB is obtained via the estimations for the total number of calls to the gap reduction procedure and the numbers of iterations performed within the critical and non-critical gap reduction procedures. To the best of our knowledge, this is the first time in the literature to establish the iteration complexity bounds for accelerated level-bundle methods to solve the constrained optimization problem, where either the objective function or constraint is smooth or weakly smooth. We provided numerical results to demonstrate the improved efficiency of ACLB in practice.

Acknowledgements This research was partially supported by NSF grants DMS-1319050, DMS-1620342, DMS-1719932, CMMI-1745382, DMS-1818886 and DMS-1925263.

References

1. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific Belmont (1999)
2. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
3. Chen, Y., Zhang, W.: Accelerated bundle level methods with inexact oracle. Sci. Sin. Math. **47**(10), 1119–1142 (2017). **(in Chinese)**
4. Chen, Y., Lan, G., Ouyang, Y., Zhang, W.: Fast bundle-level methods for unconstrained and ball-constrained convex optimization. Comput. Optim. Appl. **73**(1), 159–199 (2019)
5. Cruz, J.Y.B., de Oliveira, W.: Level bundle-like algorithms for convex optimization. J. Global Optim. **59**(4), 787–809 (2014)
6. de Oliveira, W.: Target radius methods for nonsmooth convex optimization. Oper. Res. Lett. **45**(6), 659–664 (2017)
7. de Oliveira, W., Sagastizábal, C.: Bundle methods in the XXIst century: a bird's-eye view. Pesqui. Oper. **34**(3), 647–670 (2014)
8. Fábián, C.I.: Bundle-type methods for inexact data. CEJOR **8**(1), 35–55 (2000)
9. Fábián, C.I., Wolf, C., Koberstein, A., Suhl, L.: Risk-averse optimization in two-stage stochastic models: computational aspects and a study. SIAM J. Optim. **25**(1), 28–52 (2015)
10. Jacob, L., Obozinski, G., Vert, J.-P.: Group lasso with overlap and graph lasso. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 433–440. ACM (2009)
11. Karas, E., Ribeiro, A., Sagastizábal, C., Solodov, M.: A bundle-filter method for nonsmooth convex constrained optimization. Math. Program. **116**(1), 297–320 (2009)
12. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Program. **46**(1), 105–122 (1990)
13. Kiwiel, K.C.: Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. Math. Program. **69**(1), 89–109 (1995)
14. Kiwiel, K.C.: Methods of Descent for Nondifferentiable Optimization, vol. 1133. Springer, Berlin (2006)
15. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
16. Lan, G.: Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. Math. Program. **149**(1–2), 1–45 (2015)
17. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. Math. Program. **69**(1), 111–147 (1995)
18. Mairal, J., Jenatton, R., Obozinski, G., Bach, F.: Convex and network flow optimization for structured sparsity. J. Mach. Learn. Res. **12**(Sep), 2681–2720 (2011)
19. Nemirovsky, A., Yudin, D., Dawson, E.: Problem Complexity and Method Efficiency in Optimization. Wiley, London (1983)
20. Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence o $(1/k^2)$. Dokl. SSSR **269**, 543–547 (1983)

21. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springer, Berlin (2004)
22. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
23. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springer, Berlin (2013)
24. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1), 259–268 (1992)
25. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. J. R. Stat. Soc.: Ser. B (Statistical Methodology) **67**(1), 91–108 (2005)
26. Tomioka, R., Suzuki, T., Hayashi, K., Kashima, H.: Statistical performance of convex tensor decomposition. In: Advances in Neural Information Processing Systems, pp. 972–980 (2011)
27. van Ackooij, W.: Level bundle methods for constrained convex optimization with various oracles. Comput. Optim. Appl. **57**(3), 555–597 (2014)