Math. Program., Ser. A (2015) 149:1–45
DOI 10.1007/s10107-013-0737-x

FULL LENGTH PAPER

Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization

Guanghui Lan

Received: 22 January 2013 / Accepted: 13 December 2013 / Published online: 29 December 2013

Abstract The main goal of this paper is to develop uniformly optimal first-order methods for convex programming (CP). By uniform optimality we mean that the first-order methods themselves do not require the input of any problem parameters, but can still achieve the best possible iteration complexity bounds. By incorporating a multi-step acceleration scheme into the well-known bundle-level method, we develop an accelerated bundle-level method, and show that it can achieve the optimal complexity for solving a general class of black-box CP problems without requiring the input of any smoothness information, such as, whether the problem is smooth, nonsmooth or weakly smooth, as well as the specific values of Lipschitz constant and smoothness level. We then develop a more practical, restricted memory version of this method, namely the accelerated prox-level (APL) method. We investigate the generalization of the APL method for solving certain composite CP problems and an important class of saddle-point problems recently studied by Nesterov (Math Program 103:127–152, 2005). We present promising numerical results for these new bundle-level methods applied to solve certain classes of semidefinite programming and stochastic programming problems.

Keywords Convex programming · Complexity · Bundle-level · Optimal methods

The paper is a combined version of the two manuscripts previously submitted to *Mathematical Programming*, namely: "Bundle-type methods uniformly optimal for smooth and nonsmooth convex optimization" and "Level methods uniformly optimal for composite and structured nonsmooth convex optimization".

The author of this paper was partially supported by NSF grants CMMI-1000347 and DMS-1319050, ONR grant N00014-13-1-0036, and NSF CAREER Award CMMI-1254446.

G. Lan (✉)
Department of Industrial and Systems Engineering, University of Florida,
Gainesville, FL 32611, USA
e-mail: glan@ise.ufl.edu

Mathematics Subject Classification 62L20 · 90C25 · 90C15 · 68Q25

1 Introduction

Consider the convex programming (CP)

$$f^* := \min_{x \in X} f(x), \tag{1.1}$$

where X is a convex compact set and $f : X \to \mathbb{R}$ is a closed convex function. In the classic black-box setting, f is represented by a first-order oracle which, given an input point $x \in X$, returns $f(x)$ and $f'(x) \in \partial f(x)$, where $\partial f(x)$ denotes the subdifferential of f at $x \in X$.

If f is a general nonsmooth Lipschitz continuous convex function, then, by the classic complexity theory for CP [29], the number of calls to the first-order oracle for finding an *ϵ-solution* of (1.1) (i.e., a point $\bar{x} \in X$ s.t. $f(\bar{x}) - f^* \le \epsilon$), cannot be smaller than $\mathcal{O}(1/\epsilon^2)$ when n is sufficiently large. This lower complexity bound can be achieved, for example, by the simple subgradient descent or mirror descent method [29]. If f is a smooth function with Lipschitz continuous gradient, Nesterov in a seminal work [36] presented an algorithm with the iteration complexity bounded by $\mathcal{O}(1/\epsilon^{\frac{1}{2}})$, which, by [29], is also optimal for smooth convex optimization if n is sufficiently large. Moreover, if f is a weakly smooth function with Hölder continuous gradient, i.e., $\exists$ constants $\rho \in (0, 1)$ and $M > 0$ such that $\|\nabla f(x) - \nabla f(y)\|_* \le M\|x - y\|^\rho, \forall x, y \in X$, then the optimal iteration complexity bound is given by $\mathcal{O}(1/\epsilon^{\frac{2}{1+3\rho}})$ (see [8,33,37]).

To accelerate the solutions of large-scale CP problems, much effort has recently been directed to exploiting the problem's structure, in order to identify possibly some new classes of CP problems with stronger convergence performance guarantee. One such example is given by the composite CP problems with the objective function given by $f(x) = \Psi(\phi(x))$. Here Ψ is a relatively simple nonsmooth convex function such as $\Psi(\cdot) = \|\cdot\|_1$ or $\Psi(\cdot) = \max\{y_1, \ldots, y_k\}$ (see Subsect. 4.1 for more examples) and ϕ is a k-dimensional vector function, see [9,10,19,23,30,34,35,38]. In most of these studies, the components of ϕ are assumed to be smooth convex functions. In this case, the iteration complexity can be improved to $\mathcal{O}(1/\epsilon^{\frac{1}{2}})$ by properly modifying Nesterov's optimal smooth method, see for example, [30,35,38]. It should be noted that these optimal first-order methods for general composite CP problems are in a sense "conceptual" since they require the minimization of the summation of a prox-function together with the composition of Ψ with an affine transformation [38]. More recently, Nesterov [40] studied a class of nonsmooth convex-concave saddle point problems, where the objective function f, in its basic form, is given by

$$f(x) = \max_{y \in Y} \langle Ax, y \rangle.$$

Here $Y \subseteq \mathbb{R}^m$ is a convex compact set and A denotes a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^m$. Nesterov shows that f can be closely approximated by a certain smooth convex function and that the iteration complexity for solving this class of problems can be improved to $\mathcal{O}(1/\epsilon)$. It is noted in [13] that this bound is unimprovable, for example, if Y is given by a Euclidean ball and the algorithm can only have access to A and A^* (the adjoint operator of A). These problems were later studied in [2,18,31,39,41,43] and found many interesting applications, for example, in [3,7,26].

The advantages of the aforementioned optimal first-order methods (e.g., subgradient method or Nesterov's method) mainly consist of their optimality, simplicity and cheap iteration cost. However, these methods might have some shortcomings in that each method is designed for solving a particular subclass of CP problems (e.g., smooth or nonsmooth). In particular, nonsmooth CP algorithms usually cannot make use of local smoothness properties that a nonsmooth instance might have, while it is well-known that Lipschitz continuous functions are differentiable almost everywhere within its domain. On the other hand, although it has been shown recently in [19] that Nesterov's method, which was originally designed for solving smooth CP problems, is also optimal for nonsmooth optimization when employed with a properly specified stepsize policy (see also [8] for a more recent generalization to weakly smooth CP problems), one still needs to determine some smoothness properties of f (e.g., whether f is smooth or not, i.e., $\rho = 1$ or 0, and the specific value of M), as well as some other global information (e.g., D_X and in some cases, the number of iterations N), before actually applying these generalized algorithms. Since these parameters describe the structure of CP problems over a global scope, these types of algorithms are still inherently worst-case oriented.

To address these issues, we propose to study the so-called *uniformly optimal first-order methods*. The key difference between uniformly optimal methods and existing ones is that they can achieve the best possible complexity for solving different subclasses of CP problems, but require little (preferably no) structural information for their implementation. To this end, we focus on a different type of first-order methods, namely: the bundle-level (BL) methods. Evolving from the well-known bundle methods [15,16,21], the BL method was first proposed by Lemaréchal et al. [22] in 1995. In contrast to subgradient or mirror descent methods for nonsmooth CP, the BL method can achieve the optimal $\mathcal{O}(1/\epsilon^2)$ iteration complexity for general nonsmooth CP without requiring the input of any problem parameters. Moreover, the BL method and their certain "restricted-memory" variants [4,5,44] often exhibit significantly superior practical performance to subgradient or mirror descent methods. However, to the best of our knowledge, the study on BL methods has so far been focused on general nonsmooth CP problems only.

Our contribution in this paper mainly consists of the following aspects. Firstly, we consider a general class of black-box CP problems in the form of (1.1), where f satisfies

$$f(y) - f(x) - \langle f'(x), y - x\rangle \le \frac{M}{1+\rho}\|y - x\|^{1+\rho}, \ \ \forall x, y \in X. \tag{1.2}$$

for some $M > 0$, $\rho \in [0, 1]$ and $f'(x) \in \partial f(x)$. Clearly, this class of problems cover nonsmooth ($\rho = 0$), smooth ($\rho = 1$) and weakly smooth ($\rho \in (0, 1)$) CP problems (see for example, p.22 of [38] for the standard arguments used in smooth and weakly smooth case, and Lemma 2 of [19] for a related result in the nonsmooth case). By incorporating into the BL method a multi-step acceleration scheme that was first used by Nesterov [36] and later in [2,18,19,38,40] to accelerate gradient type methods for solving smooth CP problems, we present a new BL-type algorithm, namely: the accelerated bundle-level (ABL) method. We show that the iteration complexity of the ABL method can be bounded by

$$\mathcal{O}\left\{\left(\frac{MD_X^{1+\rho}}{\epsilon}\right)^{\frac{2}{1+3\rho}}\right\}.$$

Hence, the ABL method is optimal for solving not only nonsmooth, but also smooth and weakly smooth CP problems. More importantly, this method does not require the input of any smoothness information, such as whether a problem is smooth, nonsmooth or weakly smooth, and the specific values of problem parameters M, ρ and D_X. To the best of our knowledge, this is the first time that uniformly optimal algorithms of this type have been proposed in the literature.

Secondly, one problem for the ABL method is that, as the algorithm proceeds, its subproblems become more difficult to solve. As a result, each iteration of the ABL method becomes computationally more and more expensive. To remedy this issue, we present a restricted memory version of this method, namely: the accelerated prox-level (APL) method, and demonstrate that it can also uniformly achieve the optimal complexity for solving any black-box CP problems. In particular, each iteration of the APL method requires the projection onto the feasible set X coupled with a few extra linear constraints, and the number of such linear constraints can be fully controlled (as small as 1 or 2). The basic idea of this improvement is to incorporate a novel rule due to Kiwiel [17] (later studied by Ben-tal and Nemirovski [4,5]) for updating the lower bounds and prox-centers. In addition, non-Euclidean prox-functions can be employed to make use of the geometry of the feasible set X in order to obtain (nearly) dimension-independent iteration complexity.

Thirdly, we investigate the generalization of the APL method for solving certain classes of composite and structured nonsmooth CP problems. In particular, we show that with little modification, the APL method is optimal for solving a class of generalized composite CP problems with the objective given by $f(x) = \Psi(\phi(x))$. Here $\phi_i(x), i \geq 1$, can be a mixture of smooth, nonsmooth, weakly smooth or affine components. Such a formulation covers a wide range of CP problems, including the nonsmooth, weakly smooth, smooth, minimax, and regularized CP problems (see Subsect. 4.1 for more discussions). The APL method can achieve the optimal iteration complexity for solving this class of composite problems without requiring any global information on the inner functions, such as the smoothness level and the size of Lipschitz constant. In addition, based on the APL method, we develop a completely problem-parameter free smoothing scheme, namely: the uniform smoothing level (USL) method, for solving the aforementioned class of structured CP problems with a bilinear saddle point structure [40]. We show that this method can find an ϵ-solution of these CP problems in at most $\mathcal{O}(1/\epsilon)$ iterations.

Finally, we demonstrate through our preliminary numerical experiments that these new BL type methods can be competitive and even significantly outperform existing first-order methods for solving certain classes of CP problems. Observe that each iteration of BL type methods involves the projection onto X coupled with a few linear constraints, while gradient type methods only require the projection onto X. As a result, the iteration cost of BL type methods can be higher than that of gradient type methods, especially when the projection onto X has explicit solutions. Here we would like to highlight a few interesting cases in which the application of BL type methods would be preferred: (i) the major iteration cost does not exist in the projection onto X, but the computation of first-order information (e.g., involving eigenvalue decomposition or the solutions of another optimization problem); and (ii) the projection onto X is as expensive as the projection onto X coupled with a few linear constraints, e.g., X is a general polyhedron. In particular, we show that the APL and USL methods, when applied to solving certain important classes of semidefine programming (SDP) and stochastic programming (SP) problems, can significantly outperform gradient type algorithms, as well as some existing BL type methods. The problems we tested consist of instances with up to 77, 213 decision variables.

The paper is organized as follows. In Sect. 2, we provide a brief review of the BL method and present the ABL method for black-box CP problems. We then study a restricted memory version of the ABL method, namely the APL method in Sect. 3. In Sect. 4, we investigate how to generalize the APL method for solving certain composite and structured nonsmooth CP problems. Section 5 is dedicated to the numerical experiments conducted on certain classes of SDP and SP problems. Finally, some concluding remarks are made in Sect. 6.

2 The accelerated bundle-level method

We present a new BL type method, namely: the accelerated bundle-level (ABL) method, which can uniformly achieve the optimal rate of convergence for smooth, weakly smooth and nonsmooth CP problems. More specifically, we provide a brief review of the BL method for nonsmooth minimization in Sect. 2.1, and then present the ABL method and discuss its main convergence properties in Sect. 2.2. Section 2.3 is devoted to the proof of a major convergence result used in Sect. 2.2. Throughout this section, we assume that the Euclidean space $\mathbb{R}^n$ is equipped with the standard Euclidean norm $\|\cdot\|$ associated with the inner product $\langle\cdot,\cdot\rangle$.

2.1 Review of the bundle-level method

Given a sequence of search points $x_1, x_2, \ldots, x_k \in X$, an important construct, namely, the cutting plane model, of the objective function f of problem (1.1) is given by

$$m_k(x) := \max\{h(x_i, x) : 1 \le i \le k\}, \tag{2.1}$$

where

$$h(z, x) := f(z) + \langle f'(z), x - z\rangle. \tag{2.2}$$

In the simplest cutting plane method [6,14], we approximate f by m_k and update the search points according to

$$x_{k+1} \in \text{Argmin}_{x \in X} m_k(x). \tag{2.3}$$

However, this scheme converges slowly, both theoretically and practically [29,38]. A significant progress [15,16,21] was made under the name of bundle methods (see, e.g., [11,42] for some important applications of these methods). In these methods, a prox-term is introduced into the objective function of (2.3) and the search points are updated by

$$x_{k+1} \in \text{Argmin}_{x \in X} \left\{ m_k(x) + \frac{r_k}{2} \|x - x_k^+\|^2 \right\}.$$

Here, the current prox-center x_k^+ is a certain point from $\{x_1, \ldots, x_k\}$ and r_k denotes the current penalty parameter. Moreover, the prox-center for the next iterate, i.e., x_{k+1}^+, will be set to x_{k+1} if $f(x_{k+1})$ is sufficiently smaller than $f(x_k)$. Otherwise, x_{k+1}^+ will be the same as x_k^+. The penalty r_k reduces the influence of the model m_k's inaccuracy and hence the instability of the algorithm. Note, however, that the determination of r_k usually requires certain on-line adjustments or line-search. In the closely related trust-region technique [24,45], the prox-term is put into the constraints of the subproblem instead of its objective function and the search points are then updated according to

$$x_{k+1} \in \text{Argmin}_{x \in X} \left\{ m_k(x) : \|x - x_k^+\|^2 \le R_k \right\}.$$

This approach also encounters similar difficulties for determining the size of R_k.

In an important work [22], Lemaréchal et al. introduced the idea of incorporating level sets into the bundle method. The basic scheme of their bundle-level (BL) methods consists of:

a) Update $\overline{f}^k$ to be the best objective value found so far and compute a lower bound on f^* by $\underline{f}_k = \min_{x \in X} m_k(x)$;
b) Set $l_k = \lambda \overline{f}^k + (1 - \lambda)\underline{f}_k$ for some $\lambda \in (0, 1)$;
c) Set $x_{k+1} = \text{argmin}_{x \in X} \left\{ \|x - x_k\|^2 : m_k(x) \le l_k \right\}$.

Observe that step c) ensures that the new search point x_{k+1} falls within the level set $\{x \in X : m_k(x) \le l_k\}$, while being as close as possible to x_k. We refer to x_k as the *prox-center*, since it controls the proximity between x_{k+1} and the aforementioned level set. It is shown in [22] that, if f is a general nonsmooth convex function (i.e., $\rho = 0$ in (1.2)), then the above scheme can find *an ϵ-solution of* (1.1) in at most

$$\mathcal{O}\left(C(\lambda) \frac{M^2 D_X^2}{\epsilon^2} \right) \tag{2.4}$$

iterations, where $C(\lambda)$ is a constant depending on λ and

$$D_X := \max_{x,y\in X} \|x-y\|. \tag{2.5}$$

In view of [29], the above complexity bound in (2.4) is unimprovable for nonsmooth convex optimization. Moreover, it turns out that the level sets give a stable description about the objective function f and, as a consequence, very good practical performance has been observed for the BL methods, e.g., [4,20,22].

2.2 The ABL algorithm and its main convergence properties

Based on the bundle-level method, our goal in this subsection is to present a new bundle type method, namely the ABL method, which can achieve the optimal complexity for solving any CP problems satisfying (1.2).

We introduce the following two key improvements into the classical BL methods. Firstly, rather than using a single sequence $\{x_k\}$, we employ three related sequences, i.e., $\{x_k^l\}$, $\{x_k^u\}$ and $\{x_k\}$, to build the cutting-plane models $m_k(x)$ (and hence the lower bound $\underline{f}_k$), compute the upper bounds $\overline{f}_k$, and control the proximity, respectively. Moreover, the relations among these sequences are defined carefully. In particular, we define $x_k^l = (1-\alpha_k)x_{k-1}^u + \alpha_k x_{k-1}$ and $x_k^u = (1-\alpha_k)x_{k-1}^u + \alpha_k x_k$ for a certain $\alpha_k \in (0,1]$. This type of multi-step scheme originated from the well-known Nesterov's accelerated gradient method for solving smooth CP problems [36]. Secondly, we group the iterations performed by the ABL method into different phases, and in each phase, the gap between the lower and upper bounds on f^* will be reduced by a certain constant factor. It is worth noting that, although the convergence analysis of the BL method also relies on the concept of phases (see, e.g., [4,5]), the description of this method usually does not involve phases. However, we need to use phases explicitly in the ABL method in order to define $\{\alpha_k\}$ in an optimal way to achieve the best possible complexity bounds for solving problem (1.1).

We start by describing the ABL gap reduction procedure, which, for a given search point p and lower bound lb on f^*, computes a new search point p^+ and updated lower bound lb^+ satisfying $f(p^+) - \text{lb}^+ \le \lambda\,[f(p) - \text{lb}]$ for some $\lambda \in (0,1)$.
The ABL gap reduction procedure: $(p^+, \text{lb}^+) = \mathcal{G}_{ABL}(p, \text{lb}, \lambda)$

0) Set $x_0^u = p$, $\overline{f}_0 = f(x_0^u)$, and $\underline{f}_0 = \text{lb}$. Also let $x_0 \in X$ and the cutting plane $m_0(x)$ be arbitrarily chosen, say $x_0 = x_0^u$ and $m_0(x) = h(x_0, x)$. Let $k = 1$.
1) *Update lower bound:* set $x_k^l = (1-\alpha_k)\,x_{k-1}^u + \alpha_k\, x_{k-1}$, $h(x_k^l, x) = f(x_k^l) + \langle f'(x_k^l), x - x_k^l\rangle$, $m_k(x) = \max\left\{m_{k-1}(x), h(x_k^l, x)\right\}$,

$$h_k^* = \min_{x\in X} m_k(x) \quad \text{and} \quad \underline{f}_k = \max\{\underline{f}_{k-1}, h_k^*\}; \tag{2.6}$$

2) *Update prox-center:* set $l_k = \lambda \underline{f}_k + (1-\lambda)\overline{f}_{k-1}$ and

$$x_k = \text{argmin}\left\{\|x - x_{k-1}\|^2 : m_k(x) \le l_k,\, x \in X\right\}; \tag{2.7}$$

3) *Update upper bound:* set $\overline{f}_k = \min\{\overline{f}_{k-1}, f(\alpha_k x_k + (1-\alpha_k)x^u_{k-1})\}$, and choose $x^u_k \in X$ such that $f(x^u_k) = \overline{f}_k$;
4) If $\overline{f}_k - \underline{f}_k \le \lambda(\overline{f}_0 - \underline{f}_0)$, **terminate** the procedure with $p^+ = x^u_k$ and $\text{lb}^+ = \underline{f}_k$;
5) Set $k = k + 1$ and go to Step 1.

We now add a few remarks about the above gap reduction procedure $\mathcal{G}_{ABL}$. Firstly, we say that an *iteration of procedure* $\mathcal{G}_{ABL}$ occurs whenever k increases by 1. Observe that, if $\alpha_k = 1$ for all $k \ge 1$, then an iteration of procedure $\mathcal{G}_{ABL}$ will be exactly the same as that of the BL method. In fact, in this case procedure $\mathcal{G}_{ABL}$ will reduce to one phase of the BL method as described in [4,5]. Secondly, with more general selections of $\{\alpha_k\}$, the iteration cost of procedure $\mathcal{G}_{ABL}$ is still about the same as that of the BL method. More specifically, each iteration of procedure $\mathcal{G}_{ABL}$ involves the solution of two subproblems, i.e., (2.6) and (2.7), and the computation of $f(x^l_k)$, $f'(x^l_k)$ and $f(\alpha_k x_k + (1-\alpha_k)x^u_{k-1})$, while the BL method requires the solution of two similar subproblems and the computation of $f(x_k)$ and $f'(x_k)$. Thirdly, it can be easily seen that $\underline{f}_k$ and $\overline{f}_k$, $k \ge 1$, respectively, computed by procedure $\mathcal{G}_{ABL}$ are lower and upper bounds on f^*. Indeed, by the definition of $m_k(x)$, (2.2) and the convexity of f, we have

$$m_1(x) \le m_2(x) \le \ldots m_k(x) \le f(x), \quad \forall x \in X, \tag{2.8}$$

which, in view of (2.6), then implies that $\underline{f}_1 \le \underline{f}_2 \le \ldots \underline{f}_k \le f^*$. Moreover, it follows from the definition of $\overline{f}_k$ that $\overline{f}_1 \ge \overline{f}_2 \ge \cdots \ge \overline{f}_k \ge f^*$. Hence, denoting

$$\Delta_k := \overline{f}_k - \underline{f}_k, \quad k \ge 0, \tag{2.9}$$

we have

$$\Delta_0 \ge \Delta_1 \ge \Delta_2 \ge \cdots \ge \Delta_k \ge 0. \tag{2.10}$$

By showing how Δ_k in (2.9) decreases with respect to k, we establish in Theorem 1 some important convergence properties of procedure $\mathcal{G}_{ABL}$. The proof of this result is more involved and hence provided separately in Sect. 2.3.

Theorem 1 *Let $\lambda \in (0, 1)$ and $\alpha_k \in (0, 1]$, $k = 1, 2, \ldots$, be given. Also let $\Delta_k = \overline{f}_k - \underline{f}_k$ denote the optimality gap obtained at the k-th iteration of procedure $\mathcal{G}_{ABL}$ before it terminates. Then for any $k = 1, 2, \ldots$, we have*

$$\Delta_k \le \gamma_k(\lambda) \left[(1 - \lambda\alpha_1)\Delta_0 + \frac{MD_X^{1+\rho}}{1+\rho} \|\Gamma_k(\lambda, \rho)\|_{\frac{2}{1-\rho}} \right], \tag{2.11}$$

where D_X is defined in (2.5), $\|\cdot\|_p$ is the l_p norm,

$$\gamma_k(\lambda) := \begin{cases} 1 & k = 1, \\ (1 - \lambda\alpha_k)\,\gamma_{k-1}(\lambda) & k \ge 2, \end{cases} \quad \textit{and} \tag{2.12}$$

$$\Gamma_k(\lambda, \rho) := \left\{ \gamma_1(\lambda)^{-1}\alpha_1^{1+\rho}, \gamma_2(\lambda)^{-1}\alpha_2^{1+\rho}, \ldots, \gamma_k(\lambda)^{-1}\alpha_k^{1+\rho} \right\}. \tag{2.13}$$

In particular, if λ *and* $\alpha_k \in (0,1]$, $k = 1, 2, \ldots$, *are chosen such that for some* $c_1, c_2 > 0$,

$$\gamma_k(\lambda) \le c_1 k^{-2} \quad \textit{and} \quad \gamma_k(\lambda)\|\varGamma_k(\lambda,\rho)\|_{\frac{2}{1-\rho}} \le c_2 k^{-\frac{1+3\rho}{2}}, \tag{2.14}$$

then the number of iterations performed by procedure $\mathcal{G}_{ABL}$ *can be bounded by*

$$K_{ABL}(\varDelta_0) := \left\lceil \sqrt{\frac{2c_1(1-\lambda\alpha_1)}{\lambda}} + \left(\frac{2c_2 M D_X^{1+\rho}}{(1+\rho)\varDelta_0}\right)^{\frac{2}{1+3\rho}} \right\rceil. \tag{2.15}$$

Observe that, if $\alpha_k = 1$ for all $k \ge 1$, then as mentioned before, procedure $\mathcal{G}_{ABL}$ reduces to a single phase (or segment) of the BL method and hence its termination follows by slightly modifying the standard analysis of the BL algorithm. However, such a selection of $\{\alpha_k\}$ does not satisfy the conditions stated in (2.14) and thus cannot guarantee the termination of procedure $\mathcal{G}_{ABL}$ in at most $K_{ABL}(\varDelta_0)$ iterations. Below we discuss a few possible selections of $\{\alpha_k\}$ that satisfy (2.14), in order to obtain the bound in (2.15). It should be pointed out that none of these selections rely on any problem parameters, such as M, ρ and D_X.

Proposition 1 *Let* $\gamma_k(\lambda)$ *and* $\varGamma_k(\lambda,\rho)$, *respectively, be defined in* (2.12) *and* (2.13) *for some* $\lambda \in (0,1)$ *and* $\rho \in [0,1]$.

a) If $\lambda \in (2/3, 1]$ *and* $\alpha_k = 2/[\lambda(k+2)]$, $k = 1, 2, \ldots$, *then* $\alpha_k \in (0,1)$ *and relation* (2.14) *holds with*

$$c_1 = 6 \quad \textit{and} \quad c_2 = \frac{2^{3-\rho}}{3^{\frac{1-\rho}{2}}\lambda^{-(1+\rho)}}.$$

b) If α_k, $k \ge 1$, *are recursively defined by*

$$\alpha_1 = \gamma_1 = 1, \quad \gamma_k = \alpha_k^2 = (1-\lambda\alpha_k)\gamma_{k-1}, \quad \forall k \ge 2, \tag{2.16}$$

then we have $\alpha_k \in (0,1]$ *for any* $k \ge 1$. *Moreover, condition* (2.14) *is satisfied with*

$$c_1 = 4\lambda^{-2} \quad \textit{and} \quad c_2 = \frac{4}{3^{\frac{1-\rho}{2}}}\lambda^{-\frac{1+3\rho}{2}}.$$

Proof Denoting $\gamma_k \equiv \gamma_k(\lambda)$ and $\varGamma_k \equiv \varGamma_k(\lambda,\rho)$, we first show part a). Note that by (2.12), the selection of $\{\alpha_k\}$ and the fact that $\rho \in [0,1]$, we have

$$\gamma_k = \frac{6}{(k+1)(k+2)} \quad \text{and} \quad \gamma_k^{-1}\alpha_k^{1+\rho} \le \frac{2^{1+\rho}}{6\,\lambda^{1+\rho}}(k+2)^{1-\rho}.$$

Using these relations and the simple observation $\sum_{i=1}^k i^2 = k(k+1)(2k+1)/6 \le k(k+1)^2/3$, we conclude that

$$\begin{aligned}\gamma_k \|\Gamma_k\|_{\frac{2}{1-\rho}} &\le \gamma_k \frac{2^{1+\rho}}{6\,\lambda^{1+\rho}}\left[\sum_{i=1}^k (i+2)^2\right]^{\frac{1-\rho}{2}} \le \gamma_k \frac{2^{1+\rho}}{6\,\lambda^{1+\rho}}\left[\frac{(k+2)(k+3)^2}{3}\right]^{\frac{1-\rho}{2}}\\ &= \frac{2^{1+\rho}}{3^{\frac{1-\rho}{2}}\lambda^{1+\rho}}\frac{(k+3)^{1-\rho}}{(k+1)(k+2)^{\frac{1+\rho}{2}}} \le \frac{2^{1+\rho}4^{1-\rho}}{3^{\frac{1-\rho}{2}}\lambda^{1+\rho}}k^{-\frac{1+3\rho}{2}} = \frac{2^{3-\rho}}{3^{\frac{1-\rho}{2}}\lambda^{1+\rho}}k^{-\frac{1+3\rho}{2}},\end{aligned}$$

where the last inequality follows from the facts $k+3 \le 4k$ and $k+2 \ge k+1 \ge k$ for any $k \ge 1$.

We now show that part b) holds. Note that by (2.16), we have

$$\alpha_k = \frac{1}{2}\left(-\lambda\gamma_{k-1} + \sqrt{(\lambda\gamma_{k-1})^2 + 4\gamma_{k-1}}\right), \quad k \ge 2, \tag{2.17}$$

which clearly implies that $\alpha_k > 0$, $k \ge 2$. We now show that $\alpha_k \le 1$ and $\gamma_k \le 1$ by induction. Indeed, if $\gamma_{k-1} \le 1$, then, by (2.17), we have

$$\begin{aligned}\alpha_k &\le \frac{1}{2}\left(-\lambda\gamma_{k-1} + \sqrt{(\lambda\gamma_{k-1})^2 + 4\gamma_{k-1} + 4[1-(1-\lambda)\gamma_{k-1}]}\right)\\ &= \frac{1}{2}\left(-\lambda\gamma_{k-1} + \sqrt{(\lambda\gamma_{k-1})^2 + 4\lambda\gamma_{k-1} + 4}\right) = 1.\end{aligned}$$

The previous conclusion, together with the fact that $\alpha_k^2 = \gamma_k$ due to (2.16), then also imply that $\gamma_k \le 1$. Now let us bound $1/\sqrt{\gamma_k}$ for any $k \ge 2$. First observe that, by (2.16), we have, for any $k \ge 2$,

$$\frac{1}{\sqrt{\gamma_k}} - \frac{1}{\sqrt{\gamma_{k-1}}} = \frac{\sqrt{\gamma_{k-1}} - \sqrt{\gamma_k}}{\sqrt{\gamma_{k-1}\gamma_k}} = \frac{\gamma_{k-1} - \gamma_k}{\sqrt{\gamma_{k-1}\gamma_k}\left(\sqrt{\gamma_{k-1}} + \sqrt{\gamma_k}\right)} = \frac{\lambda\alpha_k\gamma_{k-1}}{\gamma_{k-1}\sqrt{\gamma_k} + \gamma_k\sqrt{\gamma_{k-1}}}.$$

Using the above identity, (2.16) and the fact that $\gamma_k \le \gamma_{k-1}$ due to (2.16), we conclude that

$$\frac{1}{\sqrt{\gamma_k}} - \frac{1}{\sqrt{\gamma_{k-1}}} \ge \frac{\lambda\alpha_k}{2\sqrt{\gamma_k}} = \frac{\lambda}{2} \quad \text{and} \quad \frac{1}{\sqrt{\gamma_k}} - \frac{1}{\sqrt{\gamma_{k-1}}} \le \frac{\lambda\alpha_k}{\sqrt{\gamma_k}} = \lambda,$$

which, in view of the fact that $\gamma_1 = 1$, then implies that $1 + \lambda(k-1)/2 \le 1/\sqrt{\gamma_k} \le 1 + \lambda(k-1)$. Using the previous inequality and (2.16), we conclude that

$$\gamma_k \le \frac{4}{[2+\lambda(k-1)]^2} \le \frac{4}{\lambda^2 k^2}, \quad \gamma_k^{-1}\alpha_k^{1+\rho} = \left(\sqrt{\gamma_k}\right)^{-(1-\rho)} \le [1+\lambda(k-1)]^{1-\rho},$$

and

$$\gamma_k \|\Gamma_k\|_{\frac{2}{1-\rho}} \le \gamma_k \left[\sum_{i=1}^{k} [1+\lambda(i-1)]^2 \right]^{\frac{1-\rho}{2}} \le \gamma_k \left(\int_0^{2+\lambda(k-1)} u^2 \, du \right)^{\frac{1-\rho}{2}}$$
$$\le \frac{4}{3^{\frac{1-\rho}{2}}} [2+\lambda(k-1)]^{-\frac{1+3\rho}{2}} \le \frac{4}{3^{\frac{1-\rho}{2}}} (\lambda k)^{-\frac{1+3\rho}{2}}.$$

□

According to the termination criterion in step 4 of procedure $\mathcal{G}_{ABL}$, each call to this procedure will reduce the gap between a given upper and lower bound on f^* by a constant factor. In the ABL method described below, we will iteratively call procedure $\mathcal{G}_{ABL}$ until a certain accurate solution of problem (1.1) is found.

The ABL method:

Input: initial point $p_0 \in X$, tolerance $\epsilon > 0$ and algorithmic parameter $\lambda \in (0, 1)$.

0) Set $p_1 \in \text{Argmin}_{x \in X} h(p_0, x)$, $\text{lb}_1 = h(p_0, p_1)$ and $\text{ub}_1 = f(p_1)$. Let $s = 1$.
1) If $\text{ub}_s - \text{lb}_s \le \epsilon$, **terminate**;
2) Set $(p_{s+1}, \text{lb}_{s+1}) = \mathcal{G}_{ABL}(p_s, \text{lb}_s, \lambda)$ and $\text{ub}_{s+1} = f(p_{s+1})$;
3) Set $s = s + 1$ and go to step 1.

Whenever s increments by 1, we say that a phase of the ABL method occurs. Unless explicitly mentioned otherwise, an iteration of procedure $\mathcal{G}_{ABL}$ is also referred to as an iteration of the ABL method. The main convergence properties of the above ABL method are summarized as follows.

Theorem 2 *Suppose that* $\lambda \in (0, 1)$ *and* $\alpha_k \in (0, 1]$, $k = 1, 2, \ldots$, *in procedure* $\mathcal{G}_{ABL}$ *are chosen such that* (2.14) *holds for some* $c_1, c_2 > 0$. *Let* D_X, M *and* ρ *be given by* (2.5) *and* (1.2).

a) The number of phases performed by the ABL method does not exceed

$$S(\epsilon) = \left\lceil \max \left\{ 0, \log_{\frac{1}{\lambda}} \frac{M D_X^{1+\rho}}{(1+\rho)\epsilon} \right\} \right\rceil. \tag{2.18}$$

b) The total number of iterations performed by the ABL method can be bounded by

$$\left(1 + \sqrt{\frac{2c_1}{\lambda}}\right) S(\epsilon) + \frac{1}{1 - \lambda^{\frac{2}{1+3\rho}}} \left(\frac{2c_2 M D_X^{1+\rho}}{(1+\rho)\epsilon} \right)^{\frac{2}{1+3\rho}}. \tag{2.19}$$

Proof Denote $\delta_s \equiv \text{ub}_s - \text{lb}_s$, $s \ge 1$. Without loss of generality, we assume that $\delta_1 > \epsilon$, since otherwise the statements are obviously true. Note that by the origin of ub_s and lb_s, we have

$$\delta_{s+1} \le \lambda \delta_s, \quad s \ge 1. \tag{2.20}$$

Also note that, by (1.2), (2.5) and the definition of p_1 in the ABL method, we have

$$\delta_1 = f(p_1) - h(p_0, p_1) = f(p_1) - \big[f(p_0) + \langle f'(p_0), p_1 - p_0\rangle\big] \\ \le \frac{M\|p_1 - p_0\|^{1+\rho}}{1+\rho} \le \frac{MD_X^{1+\rho}}{1+\rho}. \tag{2.21}$$

The previous two observations then clearly imply that the number of phases performed by the ABL method is bounded by (2.18). We now bound the total number of iterations performed by the ABL method. Suppose that procedure $\mathcal{G}_{ABL}$ has been called $\bar{s}$ times for some $1 \le \bar{s} \le S(\epsilon)$. It then follows from (2.20) that $\delta_s > \epsilon\lambda^{s-\bar{s}}$, $s = 1, \ldots, \bar{s}$, since $\delta_{\bar{s}} > \epsilon$ due to the origin of $\bar{s}$. Using this observation, we obtain

$$\sum_{s=1}^{\bar{s}} \delta_s^{-\frac{2}{1+3\rho}} < \sum_{s=1}^{\bar{s}} \frac{\lambda^{\frac{2}{1+3\rho}(\bar{s}-s)}}{\epsilon^{\frac{2}{1+3\rho}}} = \sum_{t=0}^{\bar{s}-1} \frac{\lambda^{\frac{2}{1+3\rho}t}}{\epsilon^{\frac{2}{1+3\rho}}} \le \frac{1}{(1-\lambda^{\frac{2}{1+3\rho}})\epsilon^{\frac{2}{1+3\rho}}}.$$

Moreover, by Theorem 1, the total number of iterations performed by the ABL method is bounded by

$$\sum_{s=1}^{\bar{s}} K_{ABL}(\delta_s) \le \left(1 + \sqrt{\frac{2c_1}{\lambda}}\right)\bar{s} + \sum_{s=1}^{\bar{s}} \left(\frac{2c_2 MD_X^{1+\rho}}{(1+\rho)\delta_s}\right)^{\frac{2}{1+3\rho}}$$

Our result then immediately follows by combining the above two inequalities. □

We now add a few remarks about Theorem 2. Firstly, by setting $\rho = 0$, $\rho = 1$ and $\rho \in (0, 1)$ in (2.19), respectively, we obtain the optimal iteration complexity for nonsmooth, smooth and weakly smooth convex optimization (see [8,19,33,37] for a discussion about the lower complexity bounds for solving these CP problems). Secondly, the ABL method achieves these aforementioned optimal complexity bounds without requiring the input of any smoothness information, such as whether the problem is smooth or not, and the specific values for ρ and M in (1.2). To the best of our knowledge, the ABL method seems to be the first uniformly optimal method for solving smooth, nonsmooth and weakly smooth CP problems in the literature. Thirdly, observe that one potential problem for the ABL method is that, as the algorithm proceeds, the model $m_k(x)$ accumulates cutting planes, and the subproblems in procedure $\mathcal{G}_{ABL}$ become more difficult to solve. We will address this issue in Sect. 3 by developing a variant of the ABL method.

2.3 Convergence analysis of the ABL gap reduction procedure

Our goal in this subsection is to prove Theorem 1, which describes some important convergence properties of procedure $\mathcal{G}_{ABL}$. We will first establish three technical results from which Theorem 1 immediately follows.

Lemma 1 below shows that the prox-centers $\{x_k\}$ for procedure $\mathcal{G}_{ABL}$ are "close" to each other, in terms of $\sum_{i=1}^{k} \|x_{i-1} - x_i\|^2$. It follows the standard analysis of the BL method (see, e.g., [4,22]).

Lemma 1 *Let $m_i(x)$ and $l_i, i = 1, \ldots, k$, respectively, be computed in step 1 and step 2 of procedure $\mathcal{G}_{ABL}$ before it terminates. Then the level sets given by $\mathcal{L}_i := \{x \in X : m_i(x) \le l_i\}, i = 1, 2, \ldots, k$, have a point in common. As a consequence, we have*

$$\sum_{i=1}^{k} \|x_{i-1} - x_i\|^2 \le D_X^2, \tag{2.22}$$

where D_X is defined in (2.5).

Proof Let $\Delta_k, k = 0, 1, \ldots$, be defined in (2.9). First, in view of (2.10) and the termination criterion of procedure $\mathcal{G}_{ABL}$, we have

$$\Delta_0 \ge \Delta_1 \ge \cdots \ge \Delta_k > \lambda \Delta_0. \tag{2.23}$$

Now let $u \in \text{Argmin}_{x \in X} m_k(x)$. Observe that, by (2.6), (2.8) and (2.23), we have, for any $i = 1, 2, \ldots, k$,

$$\begin{aligned} m_i(u) &\le m_k(u) = \underline{f}_k = \overline{f}_k - \Delta_k < \overline{f}_k - \lambda \Delta_0 \le \overline{f}_i - \lambda \Delta_0 \\ &\le \overline{f}_i - \lambda \Delta_i = (1 - \lambda)\, \overline{f}_i + \lambda\, \underline{f}_i \le (1 - \lambda)\, \overline{f}_{i-1} + \lambda\, \underline{f}_i = l_i. \end{aligned}$$

We have thus shown that $u \in \mathcal{L}_i$ for any $i = 1, 2, \ldots, k$. Now by (2.7), we have

$$\|x_i - u\|^2 + \|x_{i-1} - x_i\|^2 \le \|x_{i-1} - u\|^2, \quad i = 1, 2, \ldots, k.$$

Summing up the above inequalities and using (2.5), we obtain

$$\|x_k - u\|^2 + \sum_{i=1}^{k} \|x_{i-1} - x_i\|^2 \le \|x_0 - u\|^2 \le D_X^2,$$

which clearly implies (2.22). □

The following two technical results will be used in the convergence analysis for a few accelerated bundle-level type methods, including ABL, APL and USL, developed in this paper.

Lemma 2 *Let $(x_{k-1}, x_{k-1}^u) \in X \times X$ be given at the k-th iteration, $k \ge 1$, of an iterative scheme and denote $x_k^l = \alpha_k x_{k-1} + (1 - \alpha_k) x_{k-1}^u$. Also let $h(z, \cdot)$ be defined in (2.2) and suppose that the pair of new search points $(x_k, \tilde{x}_k^u) \in X \times X$ satisfy that, for some $l \in \mathbb{R}$ and $\alpha_k \in (0, 1]$,*

$$h(x_k^l, x_k) \leq l, \tag{2.24}$$

$$\tilde{x}_k^u = \alpha_t x_k + (1-\alpha_k) x_{k-1}^u. \tag{2.25}$$

Then,

$$f(\tilde{x}_k^u) \leq (1-\alpha_k) f(x_{k-1}^u) + \alpha_k l + \frac{M}{1+\rho} \|\alpha_k (x_k - x_{k-1})\|^{1+\rho}. \tag{2.26}$$

Proof It can be easily seen from (2.25) and the definition of x_k^l that

$$\tilde{x}_k^u - x_k^l = \alpha_k (x_k - x_{k-1}). \tag{2.27}$$

Using this observation, (1.2), (2.2), (2.24), (2.25) and the convexity of f, we have

$$\begin{aligned} f(\tilde{x}_k^u) &\leq h(x_k^l, \tilde{x}_k^u) + \frac{M}{1+\rho} \|\tilde{x}_k^u - x_k^l\|^{1+\rho} && \text{(by (1.2) and (2.2))} \\ &= (1-\alpha_k) h(x_k^l, x_{k-1}^u) + \alpha_k h(x_k^l, x_k) + \frac{M}{1+\rho} \|\tilde{x}_k^u - x_k^l\|^{1+\rho} && \text{(by (2.25))} \\ &= (1-\alpha_k) h(x_k^l, x_{k-1}^u) + \alpha_k h(x_k^l, x_k) + \frac{M}{1+\rho} \|\alpha_k (x_k - x_{k-1})\|^{1+\rho} && \text{(by (2.27))} \\ &\leq (1-\alpha_k) f(x_{k-1}^u) + \alpha_k h(x_k^l, x_k) + \frac{M}{1+\rho} \|\alpha_k (x_k - x_{k-1})\|^{1+\rho} && \text{(by convexity of } f) \\ &\leq (1-\alpha_k) f(x_{k-1}^u) + \alpha_k l + \frac{M}{1+\rho} \|\alpha_k (x_k - x_{k-1})\|^{1+\rho}. && \text{(by (2.24))} \end{aligned}$$

□

Lemma 3 *Let $w_k \in (0, 1]$, $k = 1, 2, \ldots$, be given. Also let us denote*

$$W_k := \begin{cases} 1, & k = 1, \\ (1 - w_k)\, W_{k-1}, & k \geq 2. \end{cases} \tag{2.28}$$

Suppose that $W_k > 0$ for all $k \geq 2$ and that the sequence $\{\delta_k\}_{k \geq 0}$ satisfies

$$\delta_k \leq (1 - w_k)\delta_{k-1} + B_k, \quad k = 1, 2, \ldots. \tag{2.29}$$

Then, we have $\delta_k \leq W_k (1 - w_1)\delta_0 + W_k \sum_{i=1}^k (B_i / W_i)$.

Proof Dividing both sides of (2.29) by W_k, we obtain

$$\frac{\delta_1}{W_1} \leq \frac{(1-w_1)\delta_0}{W_1} + \frac{B_1}{W_1}$$

and

$$\frac{\delta_k}{W_k} \leq \frac{\delta_{k-1}}{W_{k-1}} + \frac{B_k}{W_k}, \quad \forall k \geq 2.$$

The result then immediately follows by summing up the above inequalities and rearranging the terms. □

We are now ready to provide the proof of Theorem 1.

Proof of Theorem 1 We first establish an important recursion for procedure $\mathcal{G}_{ABL}$. Let $\Delta_k = \overline{f}_k - \underline{f}_k$, $k = 1, 2, \ldots$, be the optimality gap computed at the k-th iteration of this procedure and denote $\tilde{x}_k^u \equiv \alpha_k x_k + (1-\alpha_k)x_{k-1}^u$. By the definitions of x_k^u and $\tilde{x}_k^u$, we have $f(x_k^u) \le f(\tilde{x}_k^u)$. Also by (2.7) and the definition of $m_k(x)$, we have $h(x_k^l, x_k) \le l_k$. Using these observations and Lemma 2 (with $l = l_k$), we conclude that, for any $k \ge 1$,

$$f(x_k^u) \le f(\tilde{x}_k^u) \le (1-\alpha_k) f(x_{k-1}^u) + \alpha_k l_k + \frac{M}{1+\rho}\|\alpha_k(x_k - x_{k-1})\|^{1+\rho}. \tag{2.30}$$

Subtracting $\underline{f}_k$ from both sides of the above inequality, and observing that $f(x_k^u) - \underline{f}_k = \overline{f}_k - \underline{f}_k = \Delta_k$ and $f(x_{k-1}^u) = \overline{f}_{k-1}$, we obtain

$$\Delta_k \le (1-\alpha_k)\overline{f}_{k-1} - \underline{f}_k + \alpha_k l_k + \frac{M}{1+\rho}\|\alpha_k(x_k - x_{k-1})\|^{1+\rho}, \quad \forall k \ge 1.$$

Also note that

$$\begin{aligned}
(1-\alpha_k)\overline{f}_{k-1} - \underline{f}_k + \alpha_k l_k &= (1-\alpha_k)\overline{f}_{k-1} - \underline{f}_k + \alpha_k\left[\lambda\,\underline{f}_k + (1-\lambda)\,\overline{f}_{k-1}\right] \\
&= (1-\alpha_k)\overline{f}_{k-1} + \alpha_k(1-\lambda)\,\overline{f}_{k-1} - (1-\lambda\,\alpha_k)\underline{f}_k \\
&\le (1-\alpha_k)\overline{f}_{k-1} + \alpha_k(1-\lambda)\,\overline{f}_{k-1} - (1-\lambda\,\alpha_k)\underline{f}_{k-1} \\
&= (1-\lambda\,\alpha_k)\Delta_{k-1},
\end{aligned}$$

where the inequality follows from the fact that $\underline{f}_k \ge \underline{f}_{k-1}$. Combining the above two inequalities, we arrive at

$$\Delta_k \le (1-\lambda\,\alpha_k)\Delta_{k-1} + \frac{M}{1+\rho}\|\alpha_k(x_k - x_{k-1})\|^{1+\rho}, \quad \forall\, k \ge 1. \tag{2.31}$$

Next, let $\gamma_k \equiv \gamma_k(\lambda)$ and $\Gamma_k \equiv \Gamma_k(\lambda, \rho)$ be defined in (2.12) and (2.13), respectively. Using (2.31) and Lemma 3 (with $\delta_k = \Delta_k$, $w_k = 1 - \lambda\alpha_k$, $W_k = \gamma_k$ and $B_k = M\|\alpha_k(x_k - x_{k-1})\|^{1+\rho}/(1+\rho)$), we obtain

$$\begin{aligned}
\Delta_k &\le \gamma_k(1-\lambda\alpha_1)\Delta_0 + \frac{M}{1+\rho}\sum_{i=1}^{k}\left[\gamma_i^{-1}\|\alpha_i(x_i - x_{i-1})\|^{1+\rho}\right] \\
&\le \gamma_k(1-\lambda\alpha_1)\Delta_0 + \frac{M}{1+\rho}\|\Gamma_k\|_{\frac{2}{1-\rho}}\left[\sum_{i=1}^{k}\|x_i - x_{i-1}\|^2\right]^{\frac{1+\rho}{2}},
\end{aligned} \tag{2.32}$$

where the last relation follows from Hölder's inequality. The previous conclusion, in view of (2.22), then clearly implies (2.11).

Now let us denote $K = K_{ABL}(\varDelta_0)$. It then follows from (2.11), (2.14) and (2.15) that

$$\varDelta_K \le c_1(1-\lambda\alpha_1)K^{-2}\varDelta_0 + \frac{c_2 M D_X^{1+\rho}}{1+\rho}K^{-\frac{1+3\rho}{2}} \le \frac{\lambda}{2}\varDelta_0 + \frac{\lambda}{2}\varDelta_0 = \lambda\varDelta_0,$$

and hence that procedure $\mathcal{G}_{ABL}$ will terminate in at most K iterations.

3 The accelerated prox-level method

One critical issue for the ABL method is that, as the algorithm proceeds, its subproblems, namely problem (2.6) and (2.7), accumulate constraints and thus become more and more difficult to solve. Our goal in this section is to present a variant of the ABL method, namely the accelerated prox-level method (APL), which can still uniformly achieve the optimal iteration complexity for solving smooth, weakly smooth and non-smooth CP problems, but has significantly reduced iteration cost than that of the ABL method. In addition, throughout this section, we assume that $\mathbb{R}^n \supseteq X$ is equipped with an arbitrary norm $\|\cdot\|$ (not necessarily associated with the inner product) and $\|\cdot\|_*$ denotes its conjugate. We will employ non-Euclidean prox-functions in the APL algorithm to make use of the geometry of the feasible set X, similarly to the NERML algorithm in [4,5].

We first need to introduce a new way to construct lower bounds on f^*. Let $\mathcal{E}_f(l)$ denote the level set of f given by

$$\mathcal{E}_f(l) := \{x \in X : f(x) \le l\}. \tag{3.1}$$

Also for some $z \in X$, let $h(z,x)$ be the cutting plane defined in (2.2) and denote

$$\bar{h} := \min\left\{h(z,x) : x \in \mathcal{E}_f(l)\right\}. \tag{3.2}$$

Then, it is easy to verify that

$$\min\{l, \bar{h}\} \le f(x), \quad \forall x \in X. \tag{3.3}$$

Indeed, if $l \le f^*$, then $\mathcal{E}_f(l) = \emptyset$, $\bar{h} = +\infty$ and $\min\{l,\bar{h}\} = l$. Hence (3.3) is obviously true. Now consider the case $l > f^*$. Clearly, for an arbitrary optimal solution x^* of (1.1), we have $x^* \in \mathcal{E}_f(l)$. Moreover, by (2.2), (3.2) and the convexity of f, we have $\bar{h} \le h(z,x) \le f(x)$ for any $x \in \mathcal{E}_f(l)$. Hence, $\bar{h} \le f(x^*) = f^*$ and thus (3.3) holds.

Note, however, that to solve problem (3.2) is usually as difficult as to solve the original problem (1.1). To compute a convenient lower bound of f^*, we replace $\mathcal{E}_f(l)$ in (3.2) with a convex and compact set X' satisfying

$$\mathcal{E}_f(l) \subseteq X' \subseteq X. \tag{3.4}$$

The set X' will be referred to as a *localizer* of the level set $\mathcal{E}_f(l)$. The following result shows the computation of a lower bound on f^* by solving such a relaxation of (3.2).

Lemma 4 *Let X' be a localizer of the level set $\mathcal{E}_f(l)$ for some $l \in \mathbb{R}$ and $h(z,x)$ be defined in* (2.2)*. Denote*

$$\underline{h} := \min\left\{h(z,x) : x \in X'\right\}. \tag{3.5}$$

We have

$$\min\{l, \underline{h}\} \le f(x), \quad \forall x \in X. \tag{3.6}$$

Proof Note that if $X' = \emptyset$ (i.e., (3.5) is infeasible), then $\underline{h} = +\infty$. In this case, we have $\mathcal{E}_f(l) = \emptyset$ and $f(x) \ge l$ for any $x \in X$. Now assume that $X' \ne \emptyset$. By (3.2), (3.4) and (3.5), we have $\underline{h} \le \bar{h}$, which together with (3.3), then clearly imply (3.6). □

The second new construct that we will employ in the APL method is the *prox-function* that generalizes the Euclidean distance function $\|\cdot\|^2$ used in the ABL method (see (2.7)). More specifically, consider a convex compact set $X \subseteq \mathbb{R}^n$. A function $\omega : X \to \mathbb{R}$ is called a prox-function of X with modulus σ_ω, if it is differentiable and strongly convex with modulus σ_ω, i.e.,

$$\langle \nabla\omega(x) - \nabla\omega(z), x - z\rangle \ge \sigma_\omega \|x - z\|^2, \quad \forall x, z \in X.$$

Moreover, we denote *the size of X with respect to ω* by

$$\mathcal{D}^2_{\omega,X} := \max_{x,z\in X} \{\omega(x) - \omega(z) - \langle \nabla\omega(z), x - z\rangle\}. \tag{3.7}$$

Clearly, we have

$$\|x - z\|^2 \le \frac{2}{\sigma_\omega}\mathcal{D}^2_{\omega,X} =: \Omega_{\omega,X}, \quad \forall x, z \in X. \tag{3.8}$$

Similarly to Sect. 2.2, we start by describing a new gap reduction procedure, denoted by $\mathcal{G}_{APL}$, which, for a given search point p and a lower bound lb on f^*, computes a new search point p^+ and a new lower bound lb^+ satisfying $f(p^+) - \text{lb}^+ \le q\,[f(p) - \text{lb}]$ for some $q \in (0,1)$. Note that the value of q will depend on the two algorithmic input parameters: $\beta, \theta \in (0,1)$.

The APL gap reduction procedure: $(p^+, \text{lb}^+) = \mathcal{G}_{APL}(p, \text{lb}, \beta, \theta)$

0) Set $x_0^u = p$, $\overline{f}_0 = f(x_0^u)$, $\underline{f}_0 = \text{lb}$ and $l = \beta \underline{f}_0 + (1-\beta)\overline{f}_0$. Also let $x_0 \in X$ and the initial localizer X'_0 be arbitrarily chosen, say $x_0 = p$ and $X'_0 = X$. Set the prox-function $d_\omega(x) = \omega(x) - [\omega(x_0) + \langle \omega'(x_0), x - x_0\rangle]$. Also let $k = 1$.

1) *Update lower bound:* set $x^l_k = (1-\alpha_k)x^u_{k-1} + \alpha_k x_{k-1}, h(x^l_k, x) = f(x^l_k) + \langle f'(x^l_k), x - x^l_k \rangle$,

$$\underline{h}_k := \min_{x \in X'_{k-1}} \left\{ h(x^l_k, x) \right\}, \quad \text{and} \quad \underline{f}_k := \max \left\{ \underline{f}_{k-1}, \min\{l, \underline{h}_k\} \right\}. \tag{3.9}$$

If $\underline{f}_k \ge l - \theta(l - \underline{f}_0)$, then **terminate** the procedure with $p^+ = x^u_{k-1}$ and $\text{lb}^+ = \underline{f}_k$;

2) *Update prox-center:* set

$$x_k := \text{argmin}_{x \in X'_{k-1}} \left\{ d_\omega(x) : h(x^l_k, x) \le l \right\}; \tag{3.10}$$

3) *Update upper bound:* set $\bar{f}_k = \min\{\bar{f}_{k-1}, f(\alpha_k x_k + (1-\alpha_k)x^u_{k-1})\}$, and choose x^u_k such that $f(x^u_k) = \bar{f}_k$. If $\overline{f}_k \le l + \theta(\overline{f}_0 - l)$, then **terminate** the procedure with $p^+ = x^u_k$ and $\text{lb}^+ = \underline{f}_k$;

4) *Update localizer:* choose an arbitrary X'_k such that $\underline{X}_k \subseteq X'_k \subseteq \overline{X}_k$, where

$$\underline{X}_k := \left\{ x \in X'_{k-1} : h(x^l_k, x) \le l \right\} \quad \text{and} \quad \overline{X}_k := \{ x \in X : \langle \nabla d_\omega(x_k), x - x_k \rangle \ge 0 \}; \tag{3.11}$$

6) Set $k = k+1$ and go to step 1.

We now add a few comments about procedure $\mathcal{G}_{APL}$ described above. Firstly, note that the level l used in (3.10) is fixed throughout the procedure. This is different from the ABL gap reduction procedure where the level l_k used in (2.7) changes at each iteration. Moreover, instead of having one parameter λ as in the ABL method, we need to use two parameters (i.e., β and θ) whose values are fixed a priori, say, $\beta = \theta = 0.5$.

Secondly, procedure $\mathcal{G}_{APL}$ can be terminated in either step 1 or 3. If it terminates in step 1, then we say that significant progress has been made on the lower bound $\underline{f}_k$. Otherwise, if it terminates in step 3, then significant progress has been made on the upper bound $\overline{f}_k$.

Thirdly, observe that in step 4 of procedure $\mathcal{G}_{APL}$, we can choose any set X'_k satisfying $\underline{X}_k \subseteq X'_k \subseteq \overline{X}_k$ (the simplest way is to set $X'_k = \underline{X}_k$ or $X'_k = \overline{X}_k$). While the number of constraints in $\underline{X}_k$ increases with k, the set $\overline{X}_k$ has only one more constraint than X. By choosing X'_k between these two extremes, we can control the number of constraints in subproblems (3.9) and (3.10). Hence, the iteration cost of procedure $\mathcal{G}_{APL}$ can be considerably smaller than that of procedure $\mathcal{G}_{ABL}$.

We summarize below a few more observations regarding the execution of procedure $\mathcal{G}_{APL}$.

Lemma 5 *The following statements hold for procedure $\mathcal{G}_{APL}$.*

a) $\{X'_k\}_{k \ge 0}$ *is a sequence of localizers of the level set $\mathcal{E}_f(l)$;*

b) $\underline{f}_0 \le \underline{f}_1 \le \cdots \le \underline{f}_k \le f^*$ *and* $\overline{f}_0 \ge \overline{f}_1 \ge \cdots \ge \overline{f}_k \ge f^*$ *for any* $k \ge 1$*;*

c) Problem (3.10) is always feasible unless the procedure terminates;
d) $\emptyset \neq \underline{X}_k \subseteq \overline{X}_k$ *for any* $k \geq 1$ *and hence Step 4 is always feasible unless the procedure terminates;*
e) Whenever the procedure terminates, we have $f(p^+) - \text{lb}^+ \leq q\,[f(p) - \text{lb}]$, *where*

$$q \equiv q(\beta, \theta) := 1 - (1 - \theta)\min\{\beta, 1 - \beta\}. \tag{3.12}$$

Proof We first show part a). Firstly, noting that $\mathcal{E}_f(l) \subseteq X_0'$, we can show that $\mathcal{E}_f(l) \subseteq X_k'$, $k \geq 1$, by using induction. Suppose that X_{k-1}' is a localizer of the level set $\mathcal{E}_f(l)$. Then, for any $x \in \mathcal{E}_f(l)$, we have $x \in X_{k-1}'$. Moreover, by the definition of h, we have $h(x_k^l, x) \leq f(x) \leq l$ for any $x \in \mathcal{E}_f(l)$. Using these two observations and the definition of $\underline{X}_k$ in (3.11), we have $\mathcal{E}_f(l) \subseteq \underline{X}_k$, which, in view of the fact that $\underline{X}_k \subseteq X_k'$, implies $\mathcal{E}_f(l) \subseteq X_k'$, i.e., X_k' is a localizer of $\mathcal{E}_f(l)$.

We now show part b). The first relation follows from Lemma 4, (3.9), and the fact that X_k', $k \geq 0$, are localizers of $\mathcal{E}_f(l)$ due to part a). The second relation of part b) follows immediately from the definition of $\overline{f}_k$, $k \geq 0$.

To show part c), suppose that problem (3.10) is infeasible. Then, by the definition of $\underline{h}_k$ in (3.9), we have $\underline{h}_k > l$, which implies $\underline{f}_k \geq l$, which in turn implies that the procedure should have terminated in step 1 at iteration k.

To show part d), note that by part c), the set $\underline{X}_k$ is nonempty. Moreover, by the optimality condition of (3.10) and the definition of $\underline{X}_k$ in (3.11), we have $\langle \nabla\omega(x_k), x - x_k\rangle \geq 0$ for any $x \in \underline{X}_k$, which then implies that $\underline{X}_k \subseteq \overline{X}_k$.

We now provide the proof of part e). Suppose first that the procedure terminates in step 1 of the k-th iteration. We must have $\underline{f}_k \geq l - \theta(l - \underline{f}_0)$. By using this condition, and the facts that $f(p^+) \leq \overline{f}_0$ (see part b) and $l = \beta\underline{f}_0 + (1 - \beta)\overline{f}_0$, we obtain

$$\begin{aligned} f(p^+) - \text{lb}^+ &= f(p^+) - \underline{f}_k \leq \overline{f}_0 - [l - \theta(l - \underline{f}_0)] \\ &= [1 - (1 - \beta)(1 - \theta)](\overline{f}_0 - \underline{f}_0). \end{aligned} \tag{3.13}$$

Now suppose that the procedure terminates in step 3 of the k-th iteration. We must have $\overline{f}_k \leq l + \theta(\overline{f}_0 - l)$. By using this condition, and the facts that $\text{lb}^+ \geq \underline{f}_0$ (see Lemma 5.b) and $l = \beta\underline{f}_0 + (1 - \beta)\overline{f}_0$, we have

$$f(p^+) - \text{lb}^+ = \overline{f}_k - \text{lb}^+ \leq l + \theta(\overline{f}_0 - l) - \underline{f}_0 = [1 - (1 - \theta)\beta](\overline{f}_0 - \underline{f}_0).$$

Part e) then follows by combining the above two relations. □

By showing how the gap between the upper bound (i.e., $f(x_k^u)$) and the level l decreases with respect to k, we establish in Theorem 3 some important convergence properties of procedure $\mathcal{G}_{APL}$.

Theorem 3 *Let* $\alpha_k \in (0, 1]$, $k = 1, 2, \ldots$, *be given. Also let* $(x_k^l, x_k, x_k^u) \in X \times X \times X$, $k \geq 1$, *be the search points,* l *be the level and* $d_\omega(\cdot)$ *be the prox-function in procedure* $\mathcal{G}_{APL}$. *Then, we have*

$$f(x_k^u) - l \leq (1-\alpha_1)\gamma_k(1)\,[f(x_0^u)-l] + \frac{M}{1+\rho}\left[\frac{2d_\omega(x_k)}{\sigma_\omega}\right]^{\frac{1+\rho}{2}} \gamma_k(1)\,\|\Gamma_k(1,\rho)\|_{\frac{2}{1-\rho}} \tag{3.14}$$

for any $k \geq 1$, *where* $\|\cdot\|_p$ *denotes the* l_p *norm,* $\gamma_k(\cdot)$ *and* $\Gamma_k(\cdot,\cdot)$, *respectively, are defined in* (2.12) *and* (2.13). *In particular, if* $\alpha_k \in (0,1]$, $k = 1, 2, \ldots$, *are chosen such that for some* $c > 0$,

$$\alpha_1 = 1 \quad \textit{and} \quad \gamma_k(1)\,\|\Gamma_k(1,\rho)\|_{\frac{2}{1-\rho}} \leq c\,k^{-\frac{1+3\rho}{2}}, \tag{3.15}$$

then the number of iterations performed by procedure $\mathcal{G}_{APL}$ *can be bounded by*

$$K_{APL}(\Delta_0) := \left\lceil \left(\frac{c\,M\Omega_{\omega,X}^{\frac{1+\rho}{2}}}{\beta\theta(1+\rho)\Delta_0}\right)^{\frac{2}{1+3\rho}} \right\rceil, \tag{3.16}$$

where $\Delta_0 = \overline{f}_0 - \underline{f}_0$ *and* $\Omega_{\omega,X}$ *is defined in* (3.8).

Proof We first show that the prox-centers $\{x_k\}$ in procedure $\mathcal{G}_{APL}$ are "close" to each other in terms of $\sum_{i=1}^k \|x_{i-1} - x_i\|^2$. This result slightly extends Lemma 1 for procedure $\mathcal{G}_{ABL}$. Observe that, the function $d_\omega(x)$ is strongly convex with modulus σ_ω, $x_0 = \arg\min_{x\in X} d_\omega(x)$ and $d_\omega(x_0) = 0$. Hence, we have,

$$\frac{\sigma_\omega}{2}\|x_1 - x_0\|^2 \leq d_\omega(x_1) - d_\omega(x_0) = d_\omega(x_1). \tag{3.17}$$

Moreover, by (3.11), we have $\langle \nabla d_\omega(x_k), x - x_k\rangle \geq 0$ for any $x \in \overline{X}_k$, which, together with the fact that $X_k' \subseteq \overline{X}_k$, then imply that $\langle \nabla d_\omega(x_k), x - x_k\rangle \geq 0$ for any $x \in X_k'$. Using this observation, the fact that $x_{k+1} \in X_k'$ due to (3.10), and the strong convexity of d_ω, we have

$$\begin{aligned}\frac{\sigma_\omega}{2}\|x_{k+1} - x_k\|^2 &\leq d_\omega(x_{k+1}) - d_\omega(x_k) - \langle \nabla d_\omega(x_k), x_{k+1} - x_k\rangle \\ &\leq d_\omega(x_{k+1}) - d_\omega(x_k), \quad \forall k \geq 1.\end{aligned}$$

Summing up the above inequalities with (3.17), we arrive at

$$\frac{\sigma_\omega}{2}\sum_{i=1}^k \|x_i - x_{i-1}\|^2 \leq d_\omega(x_k). \tag{3.18}$$

Next, we establish a recursion for procedure $\mathcal{G}_{APL}$. Let us denote $\tilde{x}_k^u \equiv \alpha_k x_k + (1-\alpha_k)x_{k-1}^u$, $\gamma_k \equiv \gamma_k(1)$ and $\Gamma_k \equiv \Gamma_k(1,\rho)$. By the definitions of x_k^u and $\tilde{x}_k^u$, we have $f(x_k^u) \leq f(\tilde{x}_k^u)$. Also by (3.10), we have $h(x_k^l, x) \leq l$. Using these observations and Lemma 2, we have

$$f(x_k^u) \le f(\tilde{x}_k^u) \le (1-\alpha_k) f(x_{k-1}^u) + \alpha_k l + \frac{M}{1+\rho} \|\alpha_k(x_k - x_{k-1})\|^{1+\rho}, \quad \forall k \ge 1.$$

Subtracting l from both sides of the above inequality, we obtain

$$f(x_k^u) - l \le (1-\alpha_k)[f(x_{k-1}^u) - l] + \frac{M}{1+\rho} \|\alpha_k(x_k - x_{k-1})\|^{1+\rho}, \quad \forall k \ge 1. \quad (3.19)$$

Using the above inequality and Lemma 3 (with $\delta_k = f(x_k^u) - l$, $w_k = 1-\alpha_k$, $W_k = \gamma_k$ and $B_k = M\|\alpha_k(x_k - x_{k-1})\|^{1+\rho}/(1+\rho)$), we obtain

$$\begin{aligned} f(x_k^u) - l &\le (1-\alpha_1)\gamma_k[f(x_0^u) - l] + \frac{M}{1+\rho}\gamma_k \sum_{i=1}^{k} \gamma_i^{-1} \|\alpha_i(x_i - x_{i-1})\|^{1+\rho} \\ &\le (1-\alpha_1)\gamma_k[f(x_0^u) - l] + \frac{M}{1+\rho}\|\Gamma_k\|_{\frac{2}{1-\rho}} \left[\sum_{i=1}^{k} \|x_i - x_{i-1}\|^2\right]^{\frac{1+\rho}{2}}, \quad \forall k \ge 1 \end{aligned}$$

where the last inequality follows from Hölder's inequality. The above conclusion together with (3.18) then imply that (3.14) holds.

Now, denote $K = K_{APL}(\epsilon)$ and suppose that condition (3.15) holds. Then by (3.14), (3.15), (3.7) and (3.8), we have

$$\begin{aligned} f(x_K^u) - l &\le \frac{cM}{1+\rho}\left[\frac{2d_\omega(x_K)}{\sigma_\omega}\right]^{\frac{1+\rho}{2}} K^{-\frac{1+3\rho}{2}} \le \frac{cM}{1+\rho}\Omega_{\omega,X}^{\frac{1+\rho}{2}} K^{-\frac{1+3\rho}{2}} \\ &\le \theta\beta\Delta_0 = \theta(\overline{f}_0 - l), \end{aligned}$$

where the last equality from the fact that $l = \beta\, \underline{f}_0 + (1-\beta)\, \overline{f}_0 = \overline{f}_0 - \beta\Delta_0$. Hence, procedure $\mathcal{G}_{APL}$ must terminate in step 3 of the K-th iteration. □

In view of Theorem 3, we discuss below a few possible selections of $\{\alpha_k\}$, which satisfy condition (3.15) and thus guarantee the termination of procedure $\mathcal{G}_{APL}$. It is worth noting that these selections of $\{\alpha_k\}$ do not rely on any problem parameters, including M, ρ and $\Omega_{\omega,X}$, nor on any other algorithmic parameters, such as β and θ.

Proposition 2 *Let $\gamma_k(\cdot)$ and $\Gamma_k(\cdot,\cdot)$, respectively, be defined in (2.12) and (2.13).*

a) If $\alpha_k = 2/(k+1)$, $k = 1, 2, \ldots$, then $\alpha_k \in (0, 1]$ and relation (3.15) holds with $c = 2^{1+\rho}3^{-\frac{1-\rho}{2}}$.

b) If α_k, $k = 1, 2, \ldots$, are recursively defined by

$$\alpha_1 = \gamma_1 = 1, \quad \gamma_k = \alpha_k^2 = (1-\alpha_k)\gamma_{k-1}, \quad \forall\, k \ge 2, \quad (3.20)$$

then we have $\alpha_k \in (0, 1]$ for any $k \ge 1$. Moreover, condition (3.15) is satisfied with $c = \frac{4}{3^{\frac{1-\rho}{2}}}$.

Proof We show part a) only since part b) follows directly from Proposition 1.b) with $\lambda = 1$. Denoting $\gamma_k \equiv \gamma_k(1)$ and $\Gamma_k \equiv \Gamma_k(1, \rho)$, by (2.12) and (2.13), we have

$$\gamma_k = \frac{2}{k(k+1)} \text{ and } \gamma_k^{-1}\alpha_k^{1+\rho} = \left(\frac{2}{k+1}\right)^{\rho} k \le 2^{\rho}k^{1-\rho}. \tag{3.21}$$

Using (3.21) and the simple observation that $\sum_{i=1}^{k} i^2 = k(k+1)(2k+1)/6 \le k(k+1)^2/3$, we have

$$\gamma_k\|\Gamma_k\|_{\frac{2}{1-\rho}} \le \gamma_k\left[\sum_{i=1}^{k}\left(2^{\rho}i^{1-\rho}\right)^{\frac{2}{1-\rho}}\right]^{\frac{1-\rho}{2}} = 2^{\rho}\gamma_k\left(\sum_{i=1}^{k} i^2\right)^{\frac{1-\rho}{2}} \le 2^{\rho}\gamma_k\left[\frac{k(k+1)^2}{3}\right]^{\frac{1-\rho}{2}}$$
$$= \left(2^{1+\rho}\,3^{-\frac{1-\rho}{2}}\right)\left[k^{-\frac{1+\rho}{2}}(k+1)^{-\rho}\right] \le \left(2^{1+\rho}\,3^{-\frac{1-\rho}{2}}\right) k^{-\frac{1+3\rho}{2}}.$$

□

In view of Lemma 5.e) and the termination criterion of procedure $\mathcal{G}_{APL}$, each call to this procedure can reduce the gap between a given upper and lower bound on f^* by a constant factor q (see (3.12)). In the following APL method, we will iteratively call procedure $\mathcal{G}_{APL}$ until a certain accurate solution of problem (1.1) is found.

The APL method:

Input: initial point $p_0 \in X$, tolerance $\epsilon > 0$ and algorithmic parameters $\beta, \theta \in (0, 1)$.

```
0) Set $p_1 \in \mathrm{Argmin}_{x\in X} h(p_0, x)$, $\mathrm{lb}_1 = h(p_0, p_1)$ and $\mathrm{ub}_1 = f(p_1)$. Let $s = 1$.
1) If $\mathrm{ub}_s - \mathrm{lb}_s \le \epsilon$, terminate;
2) Set $(p_{s+1}, \mathrm{lb}_{s+1}) = \mathcal{G}_{APL}(p_s, \mathrm{lb}_s, \beta, \theta)$ and $\mathrm{ub}_{s+1} = f(p_{s+1})$;
3) Set $s = s + 1$ and go to step 1.
```

Similarly to the ABL method, whenever s increments by 1, we say that a phase of the APL method occurs. Unless explicitly mentioned otherwise, an iteration of procedure $\mathcal{G}_{APL}$ is also referred to as an iteration of the APL method. The main convergence properties of the above APL method are summarized as follows.

Theorem 4 *Let M, ρ, $\Omega_{\omega,X}$ and q be defined in (1.2), (3.8) and (3.12), respectively. Suppose that $\alpha_k \in (0, 1], k = 1, 2, \ldots,$ in procedure $\mathcal{G}_{APL}$ are chosen such that condition (3.15) holds for some $c > 0$.*

a) The number of phases performed by the APL method does not exceed

$$\bar{S}(\epsilon) := \left\lceil \max\left\{0, \log_{\frac{1}{q}} \frac{M\Omega_{\omega,X}^{\frac{1+\rho}{2}}}{(1+\rho)\epsilon}\right\}\right\rceil. \tag{3.22}$$

b) The total number of iterations performed by the APL method can be bounded by

$$\bar{S}(\epsilon) + \frac{1}{1 - q^{\frac{2}{1+3\rho}}}\left(\frac{cM\Omega_{\omega,X}^{\frac{1+\rho}{2}}}{\beta\theta(1+\rho)\epsilon}\right)^{\frac{2}{1+3\rho}}. \tag{3.23}$$

Proof Denote $\delta_s \equiv \text{ub}_s - \text{lb}_s, s \geq 1$. Without loss of generality, we assume that $\delta_1 > \epsilon$, since otherwise the statements are obviously true. By Lemma 5.e) and the origin of ub_s and lb_s, we have

$$\delta_{s+1} \leq q\delta_s, \quad s \geq 1. \tag{3.24}$$

Also note that, by (2.21) and (3.8), we have

$$\delta_1 \leq \frac{M\|p_1 - p_0\|^{1+\rho}}{1+\rho} \leq \frac{M\Omega_{\omega,X}^{\frac{1+\rho}{2}}}{1+\rho}.$$

The previous two observations then clearly imply that the number of phases performed by the APL method is bounded by (3.22).

We now bound the total number of iterations performed by the APL method. Suppose that procedure $\mathcal{G}_{APL}$ has been called $\bar{s}$ times for some $1 \leq \bar{s} \leq \bar{S}(\epsilon)$. It follows from (3.24) that $\delta_s > \epsilon q^{s-\bar{s}}$, $s = 1, \ldots, \bar{s}$, since $\delta_{\bar{s}} > \epsilon$ due to the origin of $\bar{s}$. Using this observation, we obtain

$$\sum_{s=1}^{\bar{s}} \delta_s^{-\frac{2}{1+3\rho}} < \sum_{s=1}^{\bar{s}} \frac{q^{\frac{2}{1+3\rho}(\bar{s}-s)}}{\epsilon^{\frac{2}{1+3\rho}}} = \sum_{t=0}^{\bar{s}-1} \frac{q^{\frac{2}{1+3\rho}t}}{\epsilon^{\frac{2}{1+3\rho}}} \leq \frac{1}{(1-q^{\frac{2}{1+3\rho}})\epsilon^{\frac{2}{1+3\rho}}}.$$

Moreover, by Theorem 3, the total number of iterations performed by the APL method is bounded by

$$\sum_{s=1}^{\bar{s}} K_{APL}(\delta_s) \leq \bar{s} + \sum_{s=1}^{\bar{s}} \left(\frac{cM\Omega_{\omega,X}^{\frac{1+\rho}{2}}}{\beta\theta(1+\rho)\delta_s} \right)^{\frac{2}{1+3\rho}}$$

Our result then immediately follows by combining the above two inequalities. □

Clearly, in view of Theorem 4, the APL method also uniformly achieves the optimal complexity for solving smooth, weakly smooth and nonsmooth CP problems. In addition, its iteration cost can be significantly smaller than that of the APL method.

4 Level methods for solving composite and structured nonsmooth CP problems

In this subsection, we discuss two possible ways to generalize the APL method developed in Sect. 3. More specifically, we discuss a relatively easy extension of the APL method for solving an important class of composite problems in Sect. 4.1, and present a more involved generalization of this method for solving a certain class of saddle point problems in Sect. 4.2. Throughout this section, we assume that $\|\cdot\|$ is an arbitrary norm in its embedded Euclidean space (not necessarily the one associated with the inner product) and use $\|\cdot\|_2$ to denote the l_2 norm.

4.1 Composite CP problems

In this subsection, we consider the CP problem (1.1) with f given by:

$$f(x) := \Psi(\phi(x)), \tag{4.1}$$

where the outer function $\Psi : \mathbb{R}^m \to \mathbb{R}$ is Lipschitz continuous and convex, and the inner function, given by $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$, is an m-dimensional vector-function with Lipschitz continuous and convex components ϕ_i, $i = 1, \ldots, m$. For the sake of notational convenience, we refer to this class of problems as problem (1.1)–(4.1). We assume that the structure of Ψ is relatively simple in comparison with ϕ (see Examples 1-4) and known to the iterative schemes for solving (4.1), while the inner functions ϕ_i, $i = 1, \ldots, m$, are represented by the black-box first-order oracles. These first-order oracles return, given an input point $x \in X$, the function values $\phi_i(x)$ and (sub)gradients $\phi_i'(x)$. The following three additional assumptions are made about ϕ and Ψ.

Assumption 1 $\exists \rho_i \in [0, 1]$ and $M_i \geq 0$ such that:

$$\|\phi_i'(x) - \phi_i'(y)\|_* \leq M_i \|x - y\|^{\rho_i}, \quad \forall x, y \in X. \tag{4.2}$$

Observe that relation (4.2) holds with $\rho_i = 1, 0$ and $(0, 1)$, respectively, for smooth, nonsmooth and weakly smooth components ϕ_i (c.f. [8,33,37]). Clearly, if $M_i = 0$ for some $1 \leq i \leq m$, then the component ϕ_i must be affine. Otherwise, ϕ_i must be nonlinear. To fix the notation, let us assume throughout this subsection that, for a given $1 \leq m_0 \leq m$, the first m_0 components of ϕ are nonlinear, i.e., $M_i > 0$ for any $1 \leq i \leq m_0$, while the remaining $m - m_0$ components are affine, i.e., $M_i = 0$ for any $m_0 + 1 \leq i \leq m$. We make the following assumption regarding the monotonicity of Ψ with respect to these nonlinear components.

Assumption 2 The map

$$y_i \mapsto \Psi(y_1, \ldots, y_i, \ldots, y_m)$$

is monotonically nondecreasing for any $1 \leq i \leq m_0$.

In addition, we make a certain "Lipschitz-continuity" assumption about Ψ.

Assumption 3 There exists $M_0 \in [0, \infty)$ such that

$$M_0 := \sup_{y \in \mathbb{R}^m, \delta \in \mathbb{R}^m_+} \left\{ \frac{\Psi(y + \delta) - \Psi(y)}{\|\delta\|_1} : \delta_i = 0, \forall m_0 + 1 \leq i \leq m. \right\}. \tag{4.3}$$

Many CP problems can be written in the form of problem (1.1)–(4.1). We give a few interesting examples as follows.

Example 1 Nonsmooth, weakly smooth and smooth problems. Let $m = 1$ and $\Psi(y) = y$. Then, problem (1.1)–(4.1) covers the usual nonsmooth, weakly smooth and smooth CP problems, for which condition (4.2) is satisfied with $\nu = 0$, $\nu \in (0, 1)$ and $\nu = 1$, respectively.

Example 2 Minimax problems. Let $\Psi(y) = \max\{y_1, \ldots, y_m\}$. With this outer function, problem (1.1)–(4.1) becomes the minimax problem to minimize the maximum of a finite number of convex functions. It can be used, for example, to solve a system of smooth convex inequalities $\phi_i(x) \le 0$, $i = 1, \ldots, m$, where $\phi_i(x)$ are convex functions satisfying (4.2) with $\nu_1 = \cdots = \nu_m = 1$. It can also be used to solve a system of mixed smooth and nonsmooth convex inequalities if $\nu_i = 0$ or 1, $i = 1, \ldots, m$.

Example 3 Composite smooth and nonsmooth problems. Consider $\min_{x \in X} \psi(x) = \phi_1(x) + \phi_2(x)$, where ϕ_1 is a smooth component and ϕ_2 is a nonsmooth component. Clearly, we can write the problem in the form of (4.1) by setting $\phi(x) = (\phi_1(x), \phi_2(x))$ and $\Psi(y_1, y_2) = y_1 + y_2$. For this problem, we have $\rho_1 = 1$ and $\rho_2 = 0$. The applications can be found, for example, in certain penalization approaches for solving nonsmooth CP problems [19].

Example 4 Regularized problems. Consider the problem $\min_{x \in X} \phi_1(x) + \rho r(x)$, where ϕ_1 is a smooth convex function with Lipschitz continuous gradient and $r(x)$ is a continuous, nonnegative, usually nonsmooth convex function. Clearly, this problem is a special case of Example 3. However, sometimes we may want to keep the regularization term $\rho r(x)$ in the definition of Ψ, so that this term will not be linearized when defining the cutting plane model (c.f. (4.5)). For this purpose, we can put this problem in the form of (1.1)–(4.1) by setting $\phi(x) = (\phi_1(x), x) \in \mathbb{R}^{n+1}$ and $\Psi(y, x) = y + \rho r(x)$. Note that if $r(x) = \|x\|_1$ and $\phi_1(x) = \|Ax - b\|_2^2$, this problem becomes the well-known l_1 regularized least squares problem.

Since problem (1.1)–(4.1) covers nonsmooth, weakly smooth and smooth CP as certain special cases and m_0 is a given constant, in view of [8,29,33,37], a lower bound on the iteration complexity for solving this class of generalized composite problems is given by

$$\max_{i=1,\ldots,m_0} \left(\frac{M_i}{\epsilon} \right)^{\frac{2}{1+3\rho_i}}. \tag{4.4}$$

The composite CP problem described above generalizes a few other composite CP problems existing in the literature (see, for example, Nesterov [35,38], Tseng [48], Lewis and Wright [23], Sagastizźbal [46] and Nemirovski [30]). Our goal in this subsection is to show that, by properly modifying the APL method, we can uniformly achieve the optimal complexity for solving (4.1) without requiring any global information about the inner functions ϕ_i, such as, the smoothness levels ρ_i and the Lipschitz constants M_i, for all $i = 1, \ldots, m_0$.

Observing that the structure of Ψ is known, we will replace the cutting plane $h(\cdot, \cdot)$ used in procedure $\mathcal{G}_{APL}$ with the support function given by

$$h_\Psi(z, x) := \Psi(\phi(z) + \phi'(z)(x - z)), \tag{4.5}$$

where $\phi'(z)d{:=}(\langle \phi_1'(z), d\rangle; \ldots; \langle \phi_k'(z), d\rangle)$ and $\phi_i'(z) \in \partial \phi_i(z)$ for any $i = 1, \ldots, m$. We will refer to the APL method using the above support function as *the modified APL method*. Lemma 6 describes some basic properties of h_Ψ.

Lemma 6 *Let $h_\Psi(\cdot, \cdots)$ be defined in (4.5). We have*

$$h_\Psi(z, x) \le \Psi(\phi(x)) \le h_\Psi(z, x) + M_0 \sum_{i=1}^{m_0} \frac{M_i}{1+\rho_i} \|x - z\|^{1+\rho_i}, \tag{4.6}$$

where M_i, ρ_i and M_0 are given by (4.2) and (4.3), respectively.

Proof Let us denote $o_i \equiv M_i \|x - z\|^{1+\rho_i}/(1+\rho_i), i = 1, \ldots, m_0$, and $o = \big(o_1, o_2, \ldots, o_{m_0}, 0, \ldots, 0\big)$. Clearly, we have $h_\Psi(z, z) = \Psi(\phi(z)) = f(z)$ for any $z \in X$. Moreover, it follows from (1.2) and (4.2) that

$$\phi_i(z) + \langle \phi_i(z), x - z\rangle \le \phi_i(x) \le \phi_i(z) + \langle \phi_i'(z), x - z\rangle + o_i,$$

for any $i = 1, \ldots, m_0$ and that $\phi_i(x) = \phi_i(z) + \langle \nabla \phi_i(z), x - z\rangle$ for any $i = m_0 + 1, \ldots, m$. Using these observations, Assumption 2 and the definition of M_0 in (4.3), we have

$$\begin{aligned} \Psi(\phi(z) + \phi'(z)(x-z)) \le \Psi(\phi(x)) &\le \Psi(\phi(z) + \phi'(z)(x-z) + o) \\ &\le \Psi(\phi(z) + \phi'(z)(x-z)) + M_0 \sum_{i=1}^{m_0} o_i. \end{aligned}$$

□

Observe that the second relation in (4.6) depends on M_0. In view of (4.3), it can be easily seen that $M_0 = 1$ for Examples 1, 2, 3 and 4 mentioned above. We are now ready to describe the main convergence properties of the aforementioned modified APL method.

Theorem 5 *Consider the modified APL method applied to problem (1.1)–(4.1) where we replace $h(\cdot, \cdot)$ by $h_\Psi(\cdot, \cdot)$. Suppose that $\{\alpha_k\}$ in procedure $\mathcal{G}_{APL}$ are chosen such that condition (3.15) for all $\rho \in [0, 1]$ for some $c > 0$. Let $\Omega_{\omega,X}$, q, M_i, ρ_i and M_0 be defined in (3.8), (3.12), (4.2) and (4.3) respectively.*

a) The number of phases performed by the modified APL method does not exceed:

$$\mathcal{S}_\Psi(\epsilon){:=}\left\lceil \max\left\{0, \log_{\frac{1}{q}} \left(\sum_{i=1}^{m_0} \frac{M_0 M_i}{(1+\rho_i)\,\epsilon} \Omega_{\omega,X}^{\frac{1+\rho_i}{2}} \right)\right\} \right\rceil. \tag{4.7}$$

b) The total number iterations performed by the modified APL method can be bounded by

$$\mathcal{S}_\Psi(\epsilon) + \sum_{i=1}^{m_0} \left[\frac{m_0 c M_0 M_i}{\beta\theta\,(1+\rho_i)(1 - q^{\frac{2}{1+3\rho_i}})\,\epsilon} \Omega_{\omega,X}^{\frac{1+\rho_i}{2}} \right]^{\frac{2}{1+3\rho_i}}. \tag{4.8}$$

Proof The proof of the result is similar to that of Theorem 4 by noticing the difference between relations (1.2) and (4.6), and hence the details are skipped. □

We now add a few comments about the results obtained in Theorem 5. Firstly, if there exists only one nonlinear component in the inner function $\phi(\cdot)$, i.e., $m_0 = 1$, then the bound in (4.7) reduces to the bound established for the original APL method, i.e., (3.23). Moreover, for a given $m_0 > 1$, we can see from (4.4) and (4.8) that the complexity bound in (4.8) is optimal, up to a constant factor depending on m_0, for solving the composite CP problems. Finally, as shown by the following result, in certain special cases, one can improve the dependence of the iteration-complexity bound on the number of components of ϕ.

Corollary 1 *Suppose that $\rho_1 = \rho_2 = \cdots = \rho_{m_0}$ in Assumption 1. Let us denote*

$$\tilde{M} := \sup_{y \in \mathbb{R}^m, t>0} \frac{\Psi(y + t\delta_M) - \Psi(y)}{t}, \quad \delta_M := (M_1, \ldots, M_{m_0}, 0, \ldots, 0). \quad (4.9)$$

Then, the total number of phases and iterations performed by the above modified APL method applied to problem (1.1)–(4.1) can be bounded by (3.22) and (3.23), respectively, with $M = \tilde{M}$ and $\rho = \rho_1$.

Proof Similarly to Lemma 6, we can show that

$$h_\Psi(z, x) \le \Psi(\phi(x)) \le h_\Psi(z, x) + \frac{\tilde{M}}{1 + \rho_1} \|x - z\|^{1+\rho_1}.$$

The rest of the proof is similar to that of Theorem 4 and hence the details are skipped. □

Consider a special case of problem (1.1)–(4.1) where $m = m_0$, $\rho_1 = \cdots = \rho_m$ and $\Psi(y) = \max_{1 \le i \le m} y_i$ (see Example 2). We can easily see from (4.9) that $\tilde{M} = \max_{1 \le i \le m_0} M_i$ and hence that, by Corollary (1), the iteration-complexity bound of the APL method does not depend on the number of components in the inner function $\phi(\cdot)$.

4.2 Structured nonsmooth CP problems

In this subsection, we present a new BL type method for solving a class of structured nonsmooth CP problems that has recently been studied by Nesterov (c.f. [39,40]). Consider problem (1.1) with f given by

$$f(x) := \hat{f}(x) + F(x), \quad (4.10)$$

where $\hat{f} : X \to \mathbb{R}$ is a simple Lipschitz continuous convex function and

$$F(x) := \max_{y \in Y} \left\{ \langle Ax, y \rangle - \hat{g}(y) \right\}. \quad (4.11)$$

Here, $Y \subseteq \mathbb{R}^m$ is a compact convex set, $\hat{g} : Y \to \mathbb{R}$ is a continuous convex function on Y and A denotes a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^m$. Observe also that problem (1.1)–(4.10) can be written in an adjoint form:

$$\max_{y \in Y} \left\{ g(y) := -\hat{g}(y) + G(y) \right\}, \quad G(y) := \min_{x \in X} \left\{ \langle Ax, y \rangle + \hat{f}(x) \right\}. \tag{4.12}$$

While the function F given by (4.11) is a nonsmooth convex function in general, Nesterov in an important work [40] shows that it can be closely approximated by a class of smooth convex functions. We now briefly describe Nesterov's smoothing scheme as follows. Let $v(y)$ be a prox-function of Y with modulus σ_v and prox-center $c_v = \operatorname{argmin}_{y \in Y} v(y)$. Also let us denote

$$V(y) := v(y) - v(c_v) - \langle \nabla v(c_v), y - c_v \rangle,$$

and, for some $\eta > 0$,

$$F_\eta(x) := \max_y \left\{ \langle Ax, y \rangle - \hat{g}(y) - \eta \, V(y) : \; y \in Y \right\}, \tag{4.13}$$

$$f_\eta(x) := \hat{f}(x) + F_\eta(x). \tag{4.14}$$

It is shown in [40] that $F_\eta(\cdot)$ has Lipschitz-continuous gradient with constant

$$\mathcal{L}_\eta \equiv \mathcal{L}(F_\eta) := \frac{\|A\|^2}{\eta \sigma_v}, \tag{4.15}$$

where $\|A\|$ denote the operator norm of A. Moreover, the "closeness" of $F_\eta(\cdot)$ to $F(\cdot)$ depends linearly on the parameter η. In particular, we have, for every $x \in X$,

$$F_\eta(x) \le F(x) \le F_\eta(x) + \eta \, \mathcal{D}_{v,Y}, \tag{4.16}$$

and, as a consequence,

$$f_\eta(x) \le f(x) \le f_\eta(x) + \eta \, \mathcal{D}_{v,Y}, \tag{4.17}$$

where $\mathcal{D}_{v,Y}$ given by (3.7).

Nesterov shows in [40] that one can obtain an ϵ-solution of problem (1.1)–(4.10) in at most $\mathcal{O}(1/\epsilon)$ iterations, by applying a variant of his optimal smooth method [36,40] to $\min_{x \in X} f_\eta(x)$, for a properly chosen $\eta > 0$. This result is significantly better than the iteration-complexity for the black-box nonsmooth convex optimization techniques applied to (4.10). However, to implement Nesterov's approximation scheme, it is necessary to know a number of problem parameters a priori, including $\|A\|$, σ_v and $\mathcal{D}_{v,Y}$, and the total number of iterations N. To eliminate the requirement that N should be given in advance, Nesterov in [39] presented an excessive gap procedure where the above smoothing technique is applied to both the primal and dual problem (4.12). However, to apply the excessive gap procedure, one needs to know a few more parameters, including $\|A\|$, σ_v, $\mathcal{D}_{v,Y}$, σ_ω and $\mathcal{D}_{\omega,X}$, where σ_ω is the modulus of a given prox-function ω of X and $\mathcal{D}_\omega$ is defined in (3.7). In [31], Nemirovski proposed a prox-method with $\mathcal{O}(1/\epsilon)$ iteration-complexity bound for solving a slightly more general class of CP problems than (4.10). To attain the best possible iteration complexity in [31], it is still necessary to know the parameters $\|A\|$, σ_v, $\mathcal{D}_{v,Y}$, σ_ω and $\mathcal{D}_{\omega,X}$ explicitly. One possible approach for solving problem (1.1)–(4.10) would be to apply the APL method, which is shown to be optimal for smooth convex optimization, to the smooth approximation $\min_{x\in X} f_\eta(x)$ for some $\eta > 0$, similarly to Nesterov's smoothing scheme [40]. Note however, that this approach would require both the number of iterations (or the target accuracy) and the problem parameter $\mathcal{D}_v$ (see (4.17)) given a priori.

Our goal in this section is to present a completely problem parameter-free smoothing technique, namely: the *uniform smoothing level (USL) method*, obtained by properly modifying the APL method in Sect. 3. In the USL method, the smoothing parameter η is adjusted dynamically during their execution rather than being fixed in advance. Moreover, an estimate on the value of $\mathcal{D}_v$ can be provided automatically. We start by describing the USL gap reduction procedure, denoted by $\mathcal{G}_{USL}$, which will be iteratively called by the USL method. Specifically, for a given search point p, a lower bound lb on f^* and an initial estimate $\tilde{D}$ on $\mathcal{D}_{v,Y}$, procedure $\mathcal{G}_{USL}$ will either compute a new search point p^+ and a new lower bound lb^+ satisfying $f(p^+) - \text{lb}^+ \le q\,[f(p) - \text{lb}]$ for some $q \in (0,1)$, or provide an updated estimate $\tilde{D}^+$ on $\mathcal{D}_{v,Y}$ in case the current estimate $\tilde{D}$ is not accurate enough.

The USL gap reduction procedure: $(p^+, \text{lb}^+, \tilde{D}^+) = \mathcal{G}_{USL}(p, \text{lb}, \tilde{D}, \beta, \theta)$

0) Set $x_0^u = p$, $\overline{f}_0 = f(x_0^u)$, $\underline{f}_0 = \text{lb}$, $l = \beta \underline{f}_0 + (1-\beta)\overline{f}_0$, and

$$\eta := \theta(\overline{f}_0 - l)/(2\tilde{D}). \tag{4.18}$$

   Also let $x_0 \in X$ and the initial localizer X_0' be arbitrarily chosen, say $x_0 = p$ and $X_0' = X$. Set the prox-function $d(x) = \omega(x) - [\omega(x_0) + \langle \omega'(x_0), x - x_0\rangle]$. Also let $k = 1$.

1) *Update lower bound:* set $x_k^l = (1-\alpha_k)x_{k-1}^u + \alpha_k x_{k-1}$ and

$$h(x_k^l, x) = h_\eta(x_k^l, x) := \hat{f}(x) + F_\eta(x_k^l) + \langle \nabla F_\eta(x_k^l), x - x_k^l\rangle. \tag{4.19}$$

   Compute $\underline{f}_k$ according to (3.9). If $\underline{f}_k \ge l - \theta(l - \underline{f}_0)$, then **terminate** the procedure with $p^+ = x_{k-1}^u$, $\text{lb}^+ = \underline{f}_k$, and $\tilde{D}^+ = \tilde{D}$;

2) *Update prox-center:* set x_k according to (3.10);

3) *Update upper bound:* set $\bar{f}_k = \min\{\bar{f}_{k-1}, f(\alpha_k x_k + (1-\alpha_k)x_{k-1}^u)\}$, and choose x_k^u such that $f(x_k^u) = \bar{f}_k$. Check the following two possible termination criterions:

   3a) if $\overline{f}_k \le l + \theta(\overline{f}_0 - l)$, **terminate** the procedure with $p^+ = x_k^u$, $\text{lb}^+ = \underline{f}_k$ and $\tilde{D}^+ = \tilde{D}$,

   3b) Otherwise, if $f_\eta(x_k^u) \le l + \frac{\theta}{2}(\overline{f}_0 - l)$, **terminate** the procedure with $p^+ = x_k^u$, $\text{lb}^+ = \underline{f}_k$ and $\tilde{D}^+ = 2\tilde{D}$;

4) *Update localizer:* choose an arbitrary X'_k such that $\underline{X}_k \subseteq X'_k \subseteq \overline{X}_k$, where $\underline{X}_k$ and $\overline{X}_k$ are defined in (3.11);
6) Set $k = k + 1$ and go to Step 1.

We notice that there are a few essential differences between procedure $\mathcal{G}_{USL}$ described above and procedure $\mathcal{G}_{APL}$ in Sect. 3. Firstly, in comparison with procedure $\mathcal{G}_{APL}$, procedure $\mathcal{G}_{USL}$ needs to use one additional input parameter, namely $\tilde{D}$, to define η (see (4.18)) and hence the approximation function f_η in (4.14).

Secondly, we use the support functions $h_\eta(x^l_k, x)$ of $f_\eta(x)$ defined in (4.19) procedure $\mathcal{G}_{USL}$ rather than the cutting planes of $f(x)$ in procedure $\mathcal{G}_{APL}$. Notice that by (4.19), the convexity of F_η and the first relation in (4.16), we have

$$h_\eta(x^l_k, x) \le \hat{f}(x) + F_\eta(x) \le \hat{f}(x) + F(x) = f(x), \tag{4.20}$$

which implies that the functions $h_\eta(x^l_k, x)$ underestimate f everywhere on X. Hence, $\underline{f}_k$ computed in step 1 of this procedure are indeed lower bounds of f^*.

Thirdly, there are three possibles ways to terminate procedure $\mathcal{G}_{USL}$. Similarly to procedure $\mathcal{G}_{APL}$, if it terminates in step 1 and step 3a, then we say that significant progress has been made on the lower and upper bounds on f^*, respectively. The new added termination criterion in step 3b will be used only if the value of $\tilde{D}$ is not properly specified. We formalize these observations in the following simple result.

Lemma 7 *The following statements hold for procedure $\mathcal{G}_{USL}$.*

a) If the procedure terminates in step 1 or step 3a, we have $f(p^+) - \text{lb}^+ \le q[f(p) - \text{lb}]$, where q is defined in (3.12);
b) If the procedure terminates in step 3b, then $\tilde{D} < \mathcal{D}_{v,Y}$.

Proof The proof of part a) is the same as that of Lemma 5.e) and we only need to show part b). Observe that whenever step 3b occurs, we have $\overline{f}_k > l + \theta(\overline{f}_0 - l)$ and $f_\eta(x^u_k) \le l + \frac{\theta}{2}(\overline{f}_0 - l)$. Hence,

$$f(x^u_k) - f_\eta(x^u_k) = \overline{f}_k - f_\eta(x^u_k) > \frac{\theta}{2}(\overline{f}_0 - l),$$

which, in view of the second relation in (4.17), then implies that $\eta\mathcal{D}_{v,Y} > \theta(\overline{f}_0 - l)/2$. Using this observation and (4.18), we conclude that $\tilde{D} < \mathcal{D}_{v,Y}$. □

We observe that all the results in Lemma 5a–d) regarding the execution of procedure $\mathcal{G}_{APL}$ also hold for procedure $\mathcal{G}_{USL}$. In addition, similar to Theorem 3, we establish below some important convergence properties of procedure $\mathcal{G}_{USL}$ by showing how the gap between $f(x^u_k)$ and the level l decreases.

Theorem 6 *Let $\alpha_k \in (0, 1]$, $k = 1, 2, \ldots$, be given. Also let $(x^l_k, x_k, x^u_k) \in X \times X \times X$, $k \ge 1$, be the search points, l be the level and $d_\omega(\cdot)$ be the prox-function, η be the smoothing parameter (see (4.18)) in procedure $\mathcal{G}_{USL}$. Then, we have*

$$f_\eta(x^u_k) - l \le (1 - \alpha_1)\gamma_k(1)\,[f_\eta(x^u_0) - l] + \frac{\|A\|^2 d_\omega(x_k)}{\eta\sigma_\omega\sigma_v}\gamma_k(1)\,\|\Gamma_k(1, \rho)\|_\infty, \tag{4.21}$$

for any $k \geq 1$, where $\|\cdot\|_\infty$ denotes the l_∞ norm, $\gamma_k(\cdot)$ and $\Gamma_k(\cdot,\cdot)$, respectively, are defined in (2.12) *and* (2.13)*. In particular, if $\alpha_k \in (0,1]$, $k = 1, 2, \ldots$, are chosen such that condition* (3.15) *holds with $\rho = 1$ for some $c > 0$, then the number of iterations performed by procedure $\mathcal{G}_{APL}$ can be bounded by*

$$K_{USL}(\Delta_0, \tilde{D}) := \left\lceil \frac{2\|A\|}{\beta\theta\Delta_0}\sqrt{\frac{c\mathcal{D}_{\omega,X}\tilde{D}}{\sigma_\omega\sigma_v}} \right\rceil, \tag{4.22}$$

where $\mathcal{D}_{\omega,X}$ is defined in (3.7)*.*

Proof Note that, by (4.20) and (4.14), we have $h_\eta(z,x) \leq f_\eta(x)$ for any $z, x \in X$. Moreover, by (4.14), (4.19) and the fact that F_η has Lipschitz continuous gradients with constant $\mathcal{L}_\eta$, we obtain

$$\begin{aligned} f_\eta(x) - h_\eta(z,x) &= F_\eta(x) - \left[F_\eta(z) + \langle \nabla F_\eta(z), x - z\rangle\right] \\ &\leq \frac{\mathcal{L}_\eta}{2}\|x - z\|^2 = \frac{\|A\|^2}{2\eta\sigma_v}\|x - z\|^2, \end{aligned}$$

for any $z, x \in X$, where the last inequality follows from (4.15). In view of these observations, (4.21) follows from an argument similar to the one used in the proof of (3.14) with $f = f_\eta$, $M = \mathcal{L}_\eta$ and $\rho = 1$.

Now using (3.7), (3.15) (with $\rho = 1$), (4.18) and (4.21), we obtain

$$\begin{aligned} f_\eta(x_k^u) - l &\leq \frac{\|A\|^2 d_\omega(x_k)}{\eta\sigma_\omega\sigma_v}\gamma_k(1)\,\|\Gamma_k(1,\rho)\|_\infty \leq \frac{c\|A\|^2 d_\omega(x_k)}{\eta\sigma_\omega\sigma_v k^2} \\ &\leq \frac{c\|A\|^2\mathcal{D}_{\omega,X}}{\eta\sigma_\omega\sigma_v k^2} = \frac{2c\|A\|^2\mathcal{D}_{\omega,X}\tilde{D}}{\theta(\overline{f}_0 - l)\sigma_\omega\sigma_v k^2}. \end{aligned}$$

Denoting $K = K_{USL}(\Delta_0, \tilde{D})$ and noting that $\Delta_0 = \overline{f}_0 - \underline{f}_0 = (\overline{f}_0 - l)/\beta$, we conclude from the previous inequality that $f_\eta(x_K^u) - l \leq \theta(\overline{f}_0 - l)/2$. This result together with (4.17) imply that, if $\tilde{D} \geq \mathcal{D}_{v,Y}$, then $f(x_K^u) - l \leq f_\eta(x_K^u) - l + \eta\mathcal{D}_{v,Y} \leq \theta(\overline{f}_0 - l)$. In view of these two observations and the termination criterions used in step 3, procedure $\mathcal{G}_{USL}$ must terminate in at most $K_{APL}(\Delta_0, \tilde{D})$ iterations. □

In view of Lemma 7, each call to procedure $\mathcal{G}_{USL}$ can reduce the gap between a given upper and lower bound on f^* by a constant factor q, or update the estimate on $\mathcal{D}_{v,Y}$ by a factor of 2. In the following USL method, we will iteratively call procedure $\mathcal{G}_{USL}$ until a certain accurate solution is found.

The USL method:

Input: $p_0 \in X$, tolerance $\epsilon > 0$, initial estimate $Q_1 \in (0, \mathcal{D}_v]$ and algorithmic parameters $\beta, \theta \in (0,1)$.

1) Set

$$p_1 \in \operatorname{Argmin}_{x \in X}\left\{h_0(p_0, x) := \hat{f}(x) + F(p_0) + \langle F'(p_0), x - p_0\rangle\right\}, \tag{4.23}$$

$\text{lb}_1 = h_0(p_0, p_1)$ and $\text{ub}_1 := \min\{f(p_1), f(\tilde{p}_1)\}$. Let $s = 1$.

2) If $\text{ub}_s - \text{lb}_s \leq \epsilon$, **terminate**;
3) Set $(p_{s+1}, \text{lb}_{s+1}, Q_{s+1}) = \mathcal{G}_{USL}(p_s, \text{lb}_s, Q_s, \beta, \theta)$ and $\text{ub}_{s+1} = f(p_{s+1})$;
4) Set $s = s + 1$ and go to step 1.

We now make a few remarks about the USL method described above. Firstly, each phase s, $s \geq 1$, of the USL method is associated with an estimation Q_s on $\mathcal{D}_{v,Y}$, and $Q_1 \in (0, \mathcal{D}_{v,Y}]$ is a given input parameter. Note that such a Q_1 can be easily obtained by the definition of $\mathcal{D}_{v,Y}$. Secondly, we differentiate two types of phases: a phase is called *significant* if procedure $\mathcal{G}_{USL}$ terminates in step 1 or step 3a, otherwise, it is called *non-significant*. Thirdly, In view of Lemma 7.b), if phase s is non-significant, then we must have $Q_s \leq \mathcal{D}_{v,Y}$. In addition, using the previous observation, and the facts that $Q_1 \leq \mathcal{D}_{v,Y}$ and that Q_s can be increased by a factor of 2 only in the non-significant phases, we must have $Q_s \leq 2\mathcal{D}_{v,Y}$ for all significant phases.

Before establishing the complexity of the above USL method, we first present a technical result which will be used to provide a convenient estimate on the gap between the initial lower and upper bounds on f^*. The proof of this result is given in the Appendix.

Lemma 8 *Let F be defined in (4.11) and v be a prox-function of Y with modulus σ_v. We have*

$$F(x_0) - F(x_1) - \langle F'(x_1), x_0 - x_1 \rangle \leq 2 \left(\frac{2\|A\|^2 \mathcal{D}_{v,Y}}{\sigma_v} \right)^{\frac{1}{2}} \|x_0 - x_1\|, \quad \forall x_0, x_1 \in \mathbb{R}^n, \tag{4.24}$$

where $F'(x_1) \in \partial F(x_1)$ and $\mathcal{D}_{v,Y}$ is defined in (3.7).

We are now ready to show the main convergence results for the USL method.

Theorem 7 *Suppose that $\alpha_k \in (0, 1]$, $k = 1, 2, \ldots$, in procedure $\mathcal{G}_{USL}$ are chosen such that condition (3.15) holds with $\rho = 1$ for some $c > 0$. The following statements hold for the USL method applied to problem (1.1)–(4.10):*

a) the number of non-significant phases is bounded by $\tilde{\mathcal{S}}_F(Q_1) := \lceil \log \mathcal{D}_v / Q_1 \rceil$, and the number of significant phases is bounded by $\mathcal{S}_F(\epsilon) \equiv \mathcal{S}(4\bar{\Delta}_F, \epsilon, q)$, where $\mathcal{S}(\cdot, \cdot, \cdot)$ is defined in (4.7) and

$$\bar{\Delta}_F := \|A\| \sqrt{\frac{\mathcal{D}_{\omega,X} \mathcal{D}_{v,Y}}{\sigma_\omega \sigma_v}}. \tag{4.25}$$

b) the total number of gap reduction iterations performed by the USL method does not exceed

$$\mathcal{S}_F(\epsilon) + \tilde{\mathcal{S}}_F(Q_1) + \frac{\tilde{c} \bar{\Delta}_F}{\epsilon}, \tag{4.26}$$

where $\tilde{c} := 2[\sqrt{2}/(1 - q) + \sqrt{2} + 1]\sqrt{c}/\beta\theta$.

Proof Denote $\delta_s \equiv \text{ub}_s - \text{lb}_s$, $s \geq 1$. Without loss of generality, we assume that $\delta_1 > \epsilon$, since otherwise the statements are obviously true. The first claim in part a) immediately follows from the facts that a non-significant phase can occur only if $Q_1 \leq \mathcal{D}_v$ due to Lemma 7.b) and that Q_s, $s \geq 1$, is increased by a factor of 2 in each non-significant phase. In order to show the second claim in part a), we first bound the initial optimality gap $\text{ub}_1 - \text{lb}_1$. By the convexity of F, (4.14) and (4.23), we can easily see that $\text{lb}_1 \leq f^*$. Moreover, we conclude from (4.14), (4.24) and (4.23) that

$$\begin{aligned}\text{ub}_1 - \text{lb}_1 &\leq f(p_1) - \text{lb}_1 = F(p_1) - F(p_0) - \langle F'(p_0), p_1 - p_0\rangle \\ &\leq 2\left(\frac{2\|A\|^2\mathcal{D}_{v,Y}}{\sigma_v}\right)^{\frac{1}{2}} \|p_1 - p_0\| \leq 4\bar{\Delta}_F,\end{aligned}$$

where the last inequality follows from (3.8). Using this observation and Lemma 7.a), we can easily see that the number of significant phases is bounded by $\mathcal{S}_F(\epsilon)$.

We now show that part b) holds. Let $B = \{b_1, b_2, \ldots, b_k\}$ and $N = \{n_1, n_2, \ldots, n_m\}$, respectively, denote the set of indices of the significant and non-significant phases. Note that $\delta_{b_{t+1}} \leq q\,\delta_{b_t}$, $t \geq 1$, and hence that $\delta_{b_t} \geq q^{t-k}\delta_{b_k} > \epsilon q^{t-k}$, $1 \leq t \leq k$. Also observe that $Q_{b_t} \leq 2\mathcal{D}_{v,Y}$ (see the remarks right after the statement of the USL method). Using these observations and T heorem 6, we conclude that the total number of iterations performed in the significant phases is bounded by

$$\begin{aligned}\sum_{t=1}^{k} K_{USL}(\delta_{b_t}, Q_{b_t}) &\leq \sum_{t=1}^{k} K_{USL}(\epsilon q^{t-k}, 2\mathcal{D}_{v,Y}) \\ &\leq k + \frac{2\|A\|}{\beta\theta\epsilon}\sqrt{\frac{2C_1\mathcal{D}_{\omega,X}\mathcal{D}_{v,Y}}{\sigma_\omega\sigma_v}}\sum_{t=1}^{k} q^{k-t} \\ &\leq \mathcal{S}_F + \frac{2\|A\|}{\beta\theta(1-q)\epsilon}\sqrt{\frac{2C_1\mathcal{D}_{\omega,X}\mathcal{D}_{v,Y}}{\sigma_\omega\sigma_v}},\end{aligned} \tag{4.27}$$

where the last inequality follows from part a) and the observation that $\sum_{t=1}^{k} q^{k-t} \leq 1/(1-q)$. Moreover, note that $\Delta_{n_r} > \epsilon$ for any $1 \leq r \leq m$ and that $Q_{n_{r+1}} = 2Q_{n_r}$ for any $1 \leq r \leq m$. Using these observations and Theorem 6, we conclude that the total number of iterations performed in the non-significant phases is bounded by

$$\begin{aligned}\sum_{r=1}^{m} K_{USL}(\delta_{n_r}, Q_{n_r}) &\leq \sum_{r=1}^{m} K_{USL}(\epsilon, Q_{n_r}) \leq m + \frac{2\|A\|}{\beta\theta\,\epsilon}\sqrt{\frac{C_1\mathcal{D}_{\omega,X}Q_1}{\sigma_\omega\sigma_v}}\sum_{r=1}^{m} 2^{\frac{r-1}{2}} \\ &\leq \tilde{\mathcal{S}}_F + \frac{2\|A\|}{\beta\theta\,\epsilon}\sqrt{\frac{C_1\mathcal{D}_{\omega,X}Q_1}{\sigma_\omega\sigma_v}}\sum_{r=1}^{\tilde{\mathcal{S}}_F} 2^{\frac{r-1}{2}} \leq \tilde{\mathcal{S}}_F + \frac{2\|A\|}{(\sqrt{2}-1)\beta\theta\,\epsilon}\sqrt{\frac{C_1\mathcal{D}_{\omega,X}\mathcal{D}_{v,Y}}{\sigma_\omega\sigma_v}}.\end{aligned} \tag{4.28}$$

Combining (4.27) and (4.28), we obtain (4.26). □

It is interesting to observe that, if $Q_1 = \mathcal{D}_{v,Y}$, then there are no non-significant phases and the number of iterations performed by the USL method is simply bounded optimally by (4.27). In this case, we do not need to compute the value of $f_\eta(x_k^u)$ in step 3b. We refer to such a special case of the USL method as the basic smoothing level (BSL) method. It is interesting to note that, in view of Theorem 7, the USL method still achieves the optimal complexity bound in (4.26) even without a good initial estimate on $\mathcal{D}_{v,Y}$.

5 Numerical results

Our objective in this section is to report some promising results obtained for the new BL type algorithms developed in this paper. More specifically, we study in Sects. 5.1 and 5.2, respectively, the application of these methods to solve certain classes of semidefinite programming (SDP) and stochastic programming (SP) problems.

5.1 A class of SDP problems

In this subsection, we consider the classic SDP problem of

$$\min_{x \in X} \lambda_1 \left(\mathcal{A}(x)\right), \tag{5.1}$$

where $X \subseteq \mathbb{R}^n$ is a convex and compact set, $\lambda_1 : \mathbb{R}^{m \times m} \to \mathbb{R}$ denotes the maximal eigenvalue of a symmetric matrix,

$$\mathcal{A}(x) = A_0 + \sum_{i=1}^{n} x_i A_i,$$

and A_i, $i = 1, \ldots, n$, are given $m \times m$ symmetric matrices.

One can solve problem (5.1) by using interior-point methods. However, due to the high iteration cost of interior-point methods, much effort has recently been directed to the development of first-order methods for solving problem (5.1). Since problem (5.1) is in general nonsmooth, one can use general nonsmooth convex optimization methods, such as NERML (non-Euclidean restricted memory level) in [4,5] or APL in Sect. 3. In particular, Let a symmetric matrix $A \in \mathbb{R}^{m \times m}$ be given. It is well-known that the subdifferential of λ_1 at A is given by $\partial \lambda_1(A) = \text{co}\left\{uu^T : u^T u = 1, Au = \lambda_1(A)u\right\}$, where co$(\cdot)$ denotes the convex hull. Hence, λ_1 is smooth (i.e., $\partial \lambda_1(A)$ is a singleton) if and only if the maximal eigenvalue of A has multiplicity 1. In comparison with the NERML algorithm, a nice feature of the APL method is that it can automatically explore the local smoothness structures of a particular problem instance, as the objective function of (5.1) may be differentiable along certain parts of the trajectory of the algorithm. These methods, in the worst case, require $\mathcal{O}(1/\epsilon^2)$ iterations to find an ϵ-solution of problem (5.1), and the major iteration costs of these methods consist of finding a maximal eigenvector u_x of $\mathcal{A}(x)$ for a given $x \in X$ and assembling the subgradient $\mathcal{A}^* u_x$, where $\mathcal{A}^*$ denotes the adjoint operator of $\mathcal{A}$. It should be noted that other bundle type methods, such as the spectral-bundle method by Helmberg and Rendl [11], have also been developed for solving problem (5.1). The spectral-bundle method is obtained by tailoring the well-known bundle method [15,16,21] to problem (5.1). By making use of the specific structure of problem (5.1), each iteration of this method requires the solution of a quadratic semidefinite programming problem. It should also be noted that there are no complexity results available for the aforementioned spectral bundle method.

Since problem (5.1) can also be written as a bilinear saddle point problem:

$$\min_{x \in X} \left\{ \lambda_1(\mathcal{A}(x) = \max_{y \in Y} \langle \mathcal{A}(x), y \rangle \right\}, \tag{5.2}$$

where $Y := \left\{ y \in \mathbb{R}^{m \times m} | \mathrm{Tr}(y) = 1, y \succeq 0 \right\}$, we can apply Nesterov's smoothing scheme (NEST-S) [40,41] and the USL method developed in Sect. 4.2 for solving (5.2). These methods can find an ϵ-solution of (5.2) in at most $\mathcal{O}(1/\epsilon)$ iterations. It should be noted that the iteration costs of NEST-S and USL can slightly differ from each other. More specifically, USL applied to (5.2) requires a full eigenvalue decomposition and computation of the adjoint operator $\mathcal{A}^*$ to define h_η (see (4.19)) in step 1 of procedure $\mathcal{G}_{USL}$. In addition, it requires to find a maximum eigenvalue to compute $f(x_k^u)$ in step 3a of procedure $\mathcal{G}_{USL}$. On the other hand, each iteration of NEST-S requires two (or one in some variants of Nesterov's method, see, e.g., [19]) full eigenvalue decompositions and computations of the adjoint operator $\mathcal{A}^*$.

Our goal is to compare the four different algorithms, namely: NERML, APL, USL and NEST-S, applied to solve problems in the form of (5.1) or (5.2). More details about the implementation of these algorithms are as follows.

- *Prox-functions.* If the feasible set X is a standard simplex given by $\left\{ x \in \mathbb{R}^n | \sum_{i=1}^n x_i = 1, x_i \geq 0, \forall i \right\}$, the prox-function of X, as required by all these four algorithms, is set to $\omega(x) = \sum_{i=1}^n x_i \log x_i$ and the norm is set to $\|x\|_1$. If X is a box, then we set $\omega(x) = \|x\|_2^2/2$ and the norm is set to $\|x\|_2$. The prox-function of Y, as required by the algorithms USL and NEST-S, is set to $v(y) = \sum_{i=1}^n \lambda_i(y) \log \lambda_i(y)$, and the norm is set to $\sum_{i=1}^n \lambda_i(y)$, where $\lambda_i(y), i = 1, \ldots, n$, denote the eigenvalues of $y \in Y$. Under this setting, the value of $\mathcal{D}_{v,Y}$ can be bounded by $\ln m$. Hence, we can set $Q_1 = \ln m$ in our implementation of the USL method.
- *Localizers.* For the APL, USL and NERML algorithms, we define the localizer X_k' as

  $$X_k' = \left\{ x \in X : \langle \nabla \omega(x_k^l), x - x_k \rangle \geq 0 \right\} \bigcap \mathcal{M}_k, \quad k \geq 1,$$

  where $\mathcal{M}_k$ denotes the intersection of totally at most B half spaces of the form $\{x : h(x_k^l, x) \leq l\}$ which have been generated most recently. Note that the larger the value of B is, the more difficult the subproblems of BL type methods (e.g., (3.9) and (3.10)) are. On the other hand, a larger B might help to compute a better lower bound in (3.9). It is found from our initial experiments that different values of B within [10, 30] perform almost equally well.

- *Subproblems for BL type methods.* If the problem dimension n is relatively small, say $n \leq 5000$, we can solve the subproblems for BL type methods (e.g., (3.9) and (3.10)) by Mosek [28]. In this case we set the bundle limit B to 30. If the size of n is very big, it will be time-consuming to directly solve the subproblems of BL type methods. However, observing that the number of constraints in these subproblems is very small (at most $B + 1$), one can conveniently solve the Lagrangian duals of these subproblems (see Ben-tal and Nemirovski [5]). In particular, if n is big, say $n \geq 5000$, we set $B = 10$ and solve the Lagrangian dual of these subproblems by using the ABL method, which can solve efficiently small dimensional CP problems, similarly to the BL method [5,22].
- *Fine-tuning for NEST-S.* For the NEST-S scheme, we compute the Lipschitz constant $\mathcal{L}_\eta$ by (4.15), where the smoothing parameter η is set to

  $$\frac{2\|\mathcal{A}\|}{N+1}\frac{D_{\omega,X}}{\sigma_\omega \sigma_\upsilon D_{\upsilon,Y}}.$$

  and the operator norm $\|\mathcal{A}\|$ is computed according to [41]. Note however, that the resulting estimate of $\mathcal{L}_\eta$ can be rather conservative, which leads to the slow convergence of the NEST-S scheme. We had also implemented a variant of Nesterov's method which can adaptively search for the Lipschitz constant $\mathcal{L}_\eta$ ([35]). However, our preliminary experiments indicate that the improvement from this approach is not significant. In our final experiments, we run NEST-S four times, and each time we multiply the Lipschitz constant $\mathcal{L}_\eta$ estimated above by a different factor: 10^{-1}, 10^{-2}, 10^{-3}, or 10^{-4}. We then report the best solutions, in terms of the objective value, obtained from these four runs of the NEST-S scheme.
- *Others.* We set $\beta = \theta = 1/2$, and specify $\{\alpha_k\}$ according to Proposition 2.a) in the APL and USL methods. All the codes are implemented in MATLAB2007 under Windows Vista and the experiments were conducted on an INTEL 2.53 GHz labtop.

Our first experiments were conducted on a set of randomly generated SDP instances, each of which has various sizes of A_i's for $i = 1, \ldots, n$. We also assume that the feasible set X is given by a standard simplex. More details about these instances are shown in Table 1, where n is the dimension of x, ub_1 is the objective value at $p_1 = (1/n, \ldots, 1/n)$, and $\Delta_1 = \text{ub}_1 - \text{lb}_1$ denotes the initial gap with lb_1 given by (4.23). We run 200 iterations for the four algorithms mentioned above and report the objective values obtained at the 100th and 200th iteration in column 2 and column 4, respectively. For the APL, USL and NERML algorithms, we also report the optimality gap Δ_s at the 100th and 200th iteration, respectively, in column 3 and 5 of Table 1. The CPU time (in seconds) for running these algorithms is reported in column 6 of Table 1. It should be noted that we only report the CPU time for one run of the NEST-S algorithm, although we had run it for 4 times to find a good estimate of $\mathcal{L}_\eta$.

We can draw a few conclusions from our experiments with these random SDP instances. Firstly, among the two methods with $\mathcal{O}(1/\epsilon)$ convergence, USL can significantly outperform NEST-S: the former algorithm can reach 6 accuracy digits after 200 iterations while the latter algorithm reaches at most 2 accuracy digits for these

Table 1 Experiments with random SDP problems

alg.	ub_{100}	Δ_{100}	ub_{200}	Δ_{200}	Time
E1: $n = 1{,}000$, $m = 400$, $d = 2\,\%$, $\text{ub}_1 = 6.329960$, $\Delta_1 = 5.08e-1$					
USL	6.026060	$8.48e-5$	6.026045	$1.28e-6$	174.10
NEST-S	6.077040	–	6.076351	–	190.51
APL	6.026048	$4.23e-5$	6.026045	$1.22e-6$	109.59
NERML	6.026323	$7.97e-4$	6.026084	$2.39e-4$	100.37
E2: $n = 1{,}000$, $m = 600$, $d = 2\,\%$, $\text{ub}_1 = 7.788735$, $\Delta_1 = 6.45e-1$					
USL	7.458582	$3.39e-4$	7.458538	$5.07e-6$	364.93
NEST-S	7.539811	–	7.538583	–	552.12
APL	7.458561	$8.96e-5$	7.458537	$1.96e-6$	166.27
NERML	7.458801	$1.07e-3$	7.458557	$8.35e-5$	142.16
E3: $n = 1{,}000$, $m = 800$, $d = 2\,\%$, $\text{ub}_1 = 8.855385$, $\Delta_1 = 5.57e-1$					
USL	8.555496	$1.35e-4$	8.555475	$4.08e-6$	799.30
NEST-S	8.635632	–	8.635473	–	1347.26
APL	8.555484	$6.61e-5$	8.555475	$2.05e-6$	275.22
NERML	8.555743	$7.45e-4$	8.555494	$1.24e-4$	213.25

instances. Secondly, for the two nonsmooth methods, APL consistently outperforms NERML in solution quality while the computational time is comparable to the latter one. Thirdly, while the solution quality of the USL method is significantly better than the one of the NERML algorithm, it is interesting to notice that the solution quality of the APL algorithm is comparable or better than that of the USL. One plausible explanation is that the problems to be solved, due to the inherent randomness, are smooth along most part of the trajectory of the APL algorithm.

Our second experiments were carried out for a class of more structured SDP instances, namely a class of Lovasz capacity problems. Let (N, E) denote a graph with m nodes in N and n edges in E. The Lovasz capacity ϑ of (N, E) is defined by

$$\vartheta := \min_{x \in X} \{\Phi(x) := \lambda_1\,(d + x)\}. \tag{5.3}$$

Here $X := \left\{x \in \mathcal{S}^m : x_{ij} = 0 \text{ if } (i, j) \notin E\right\}$, $\mathcal{S}^m$ denotes the set of symmetric matrices in $\mathbb{R}^{m \times m}$ and d is a $m \times m$ constant matrix given by

$$d_{ij} := \begin{cases} 0, & (i, j) \in E, \\ 1, & (i, j) \notin E. \end{cases}$$

Note that for an optimal x of problem (5.3) the matrix $\vartheta I - (d + x)$ is positive semidefinite, so that nonzero entries in x satisfy $|x_{ij}| \le \vartheta - 1$. It follows that if v is a valid upper bound on ϑ, then problem (5.3) is equivalent to

$$\vartheta = \min_{x \in X_v} \Phi(x), \tag{5.4}$$

Table 2 Lovasz capacity instances

Inst.	m	n	$\Phi(0)$
G41	400	11,503	342.62
G42	400	21,692	291.78
G51	500	23,076	407.88
G52	500	47,910	308.72
G61	600	51,429	428.87
G62	600	77,213	343.10

Table 3 Comparison of first-order methods for Lovasz capacity instances

Inst.	NERML			APL			USL			NEST-S		
	Iter.	ub	Time	Iter.	ub	Time	Iter.	ub	Time	Iter.	ub	Time
G41	1,000	63.88	829.58	800	63.61	890.18	40	62.79	44.08	1,000	342.31	980.29
G42	1,000	41.26	963.82	1,000	41.33	1166.17	30	41.19	37.89	1,000	291.43	1078.53
G51	1,000	62.82	1292.38	1,000	63.21	1697.79	30	61.13	54.56	1,000	407.68	1964.73
G52	1,000	42.19	1724.29	70	41.50	166.54	20	38.72	45.30	1,000	308.46	2403.18
G61	1,000	68.53	2343.86	20	66.76	60.46	10	59.30	32.67	1,000	428.72	3948.19
G62	1,000	40.43	4062.83	110	40.36	455.54	20	39.47	96.89	1,000	342.91	4309.73

where $X_v := \{x \in X : |x_{ij}| \le v - 1\}$. In view of this observation, we incorporate following enhancement into the aforementioned first-order methods (NERML, APL, USL and NEST-S) applied to (5.4): for all these methods, we update the upper bound v in defining the feasible set X_v from time to time. In particular, we update the value of v in each phase of BL type methods. For NEST-S, we update the value of v whenever a new upper bound on ϑ becomes available (as noted by [48], the optimal convergence of Nesterov's method will be guaranteed with such a domain shrinking strategy).

We generate a set of random graph instances as follows. For a given number of nodes m and a designed number of edges $\bar{n}$, we first generate $m - 1$ edges, each one connecting a new node with a randomly selected existing node. After that, we create $\bar{n} - m + 1$ random edges and remove those redundant edges. Thus, the actual number of edges n can be smaller than the designed number $\bar{n}$. Totally 6 instances have been generated in this manner and the number of edges n (and hence the number of decision variables) ranges from $11,503$ to $77,213$ (see Table 2). We also report the initial objective values of these instances at $x_0 = 0$ in column 4 of Table 2. In order to compare the aforementioned algorithms for computing Lovasz capacity, we first run the NERML algorithm for $1,000$ iterations and record the quality of the output solutions in terms of the generated upper bound. We then terminate the remaining three algorithms, namely APL, USL and NEST-S, whenever similar solution quality is achieved or the $1,000$ iteration limit is reached. We report the number of iterations, the computed upper bound and CPU time in columns $2-4$, $5-7$, $8-9$ and $10-12$, respectively, for NERML, APL, USL and NEST-S in Table 3. From these results, we can safely draw the following conclusions. Firstly, all the BL type methods significantly outperform NEST-S for these Lovasz capacity instances. Secondly, while

APL significantly outperforms NERML for the bigger instances G52, G61 and G62, the USL method, which combines the advantages of both APL and NEST-S, can significantly outperform NEST-S, NERML and APL for all these Lovasz capacity instances.

5.2 A class of two-stage stochastic programming problems

In this subsection, we consider the classic two-stage stochastic linear programming given by

$$\min_{x \in X} \left\{ f(x) = c^T x + \mathbb{E}[V(x, \xi)] \right\}, \tag{5.5}$$

with

$$V(x, \xi) = \min \left\{ q^T \pi : W\pi = h + Tx, \pi \geq 0 \right\}. \tag{5.6}$$

Here, $x \in \mathbb{R}^{n_1}$ and $\pi \in \mathbb{R}^{n_2}$, respectively, are the first and second-stage decision variables, $X \subseteq \mathbb{R}^{n_1}$ is a nonempty convex compact set, and $\xi \equiv (q, h, T)$ is a random vector with a known distribution supported on $\Xi \subseteq \mathbb{R}^{n_2+m_2+m_2\times n_1}$. We assume that problem (5.6) is feasible for every possible realization of ξ, i.e., problem (5.5) has a complete recourse. Moreover, for the purpose of illustrating the effectiveness of the algorithms developed in this paper, we assume that ξ is a discrete random vector and the number of possible realizations of ξ (or the sample space) is not too big.

It should be noted that if ξ is a continuous random vector or the number of possible realizations of ξ is astronomically large, to solve problem (5.5) is highly challenging, due to the fundamental difficulty of computing the expectation to a high accuracy when the dimension of ξ is high, see [20,32] for a discussion on some recent advancements in this area. However, if the number of possible realizations of ξ is not astronomically large, it is possible to solve problem (5.5) to high accuracy in a reasonable amount of time by using more powerful algorithms. This is indeed what we intend to demonstrate in this subsection.

Since problem (5.5) is nonsmooth in general, one can apply the NERML or APL methods. These methods, in the worst case, require $\mathcal{O}(1/\epsilon^2)$ iterations to find an ϵ-solution of problem (5.5). Recently, Ahmed [1] noted that one can improve the complexity bound for solving (5.5) to $\mathcal{O}(1/\epsilon)$, by applying Nesterov's smoothing scheme to (5.5). The basic idea is as follows. Let $\mathcal{Y}(q) := \{W^T y \leq q\}$ and $\mathcal{B}_{m_2}$ be the Euclidean ball in $\mathbb{R}^{m_2}$. Note that by strong duality, we have

$$V(x, \xi) = \max \left\{ (h + Tx)^T y : y \in \mathcal{Y}(q) \right\}. \tag{5.7}$$

Moreover, by Hoffman's Lemma [12], there exists a constant $\mathcal{R}_W > 0$ depending on W such that

$$\mathcal{Y}(q) \subseteq \mathcal{Y}(0) + \mathcal{R}_W \|q\| \mathcal{B}_{m_2} = \mathcal{R}_W \|q\| \mathcal{B}_{m_2},$$

Table 4 Dimension of the SP instances

	n_1	m_1	n_2	m_2
SSN	89	1	796	175
20-term	63	3	764	124

where the last identity follows from the fact that $\mathcal{Y}(0) = \{0\}$ due to the complete recourse assumption. In other words, the feasible region of (5.7) is bounded. We can then uniformly approximate $f(x)$ in (5.5) by $f_\eta(x) := c^T x + \mathbb{E}[V_\eta(x, \xi)]$ for some $\eta > 0$, where

$$V_\eta(x, \xi) = \max\left\{(h + Tx)^T y - \eta\|y\|^2/2 : y \in \mathcal{Y}(q)\right\}. \tag{5.8}$$

However, the implementation of Nesterov's smoothing scheme is difficult, since it is necessary to fine-tune a large number of problem parameters, including $\mathcal{R}_W$, $\|q\|$ and $\|T\|$, as well as $\mathcal{D}_{\omega,X}$. Due to the lack of good estimations for these parameters, especially, $\mathcal{R}_W$, no computational results have been reported in [1].

In our experiments, we have implemented three methods, namely: APL and NERML and USL, applied to problem (5.5). All these methods do not require the input of any problem parameters and the implementation details are similar to those in Subsect. 5.1. We conduct our experiments on a few SP instances which have been studied by a few authors, namely: a telecommunication design (SSN) problem of Sen, Doverspike, and Cosares [47] and the motor freight carrier routing problem (20-term) of Mak, Morton, and Wood [27]. The dimensions of these instances are shown in Table 4, please see [25] for more details about these instances.

It is worth noting that here we assume that the number of possible realizations are fixed ($N = 50$ or 100) and hence obtain four different instances, namely: SSN(50), SSN(100), 20-term(50) and 20-term(100). Noting that the initial optimality gap for these instances are rather high (in order of 10^3 or 10^7), we run each algorithm for 400 iterations and the results are reported in Table 5. The structure of the table is similar to Table 1 (see Subsect. 5.1). We also run NERML for 1, 000 iterations first and then check whether APL and USL can achieve similar gap reduction. The latter results are reported in Table 6.

We can make a few observations from the numerical results in Tables 5 and 6. Firstly, the iteration cost of the USL method is larger than that of the APL method, which, in turn, is larger than that of the NERML algorithm. In particular, the major iteration cost of the NERML and APL algorithm consists of solving N and $2N$ second-stage LP problems respectively, while the one of the USL algorithm involves the solutions of N smoothed quadratic programming problems (see (5.8)). Secondly, both the APL and USL methods can significantly outperform the the NERML algorithm in terms of the solution quality. As we can see from Tables 5 and 6, the NERML algorithm makes little progresses after 200 iterations for these SP instances. Thirdly, the solution quality of the APL method is worse than that of the USL method for solving the first two instances: SSN(50) and SSN(100), but it significantly outperforms the latter one for solving the last two instances: 20-term(50) and 20-term(100). One possible reason is that the sizes of $\mathcal{D}_{v,Y} (\approx \mathcal{R}_W^2\|q\|^2)$ for the last two instances are significantly larger

Table 5 Experiments with the SP instances

alg.	ub_{200}	Δ_{200}	ub_{400}	Δ_{400}	Time
SSN(50): $\text{ub}_1 = 2.352586e+2$, $\Delta_1 = 3.923265\text{e}+3$					
APL	4.839075	$2.437395e-3$	4.838074	$5.053628e-7$	366.30
USL	4.838125	$1.837968e-4$	4.838073	$6.599449e-7$	754.31
NERML	5.550903	$5.012402e+0$	5.086603	$1.303485e+0$	193.47
SSN(100): $\text{ub}_1 = 2.407279e+2$, $\Delta_1 = 4.023982\text{e}+3$					
APL	7.354770	$9.148243e-3$	7.352610	$4.198017e-6$	730.15
USL	7.354090	$2.683424e-3$	7.352610	$6.804606e-7$	1471.62
NERML	8.323381	$4.771295e+0$	7.578802	$1.079491e+0$	383.27
20-term(50): $\text{ub}_1 = 7.718543e+5$, $\Delta_1 = 1.804693\text{e}+7$					
APL	$2.549453e+5$	$1.229655e-3$	$2.549453e+5$	$2.405432e-7$	1056.82
USL	$2.551031e+5$	$1.896133e+3$	$2.549602e+5$	$3.310795e+2$	1209.53
NERML	$2.587140e+5$	$1.473815e+4$	$2.576649e+5$	$1.368899e+4$	301.03
20-term(100): $\text{ub}_1 = 7.664067e+5$, $\Delta_1 = 1.801832e+7$					
APL	$2.532875e+5$	$3.679608e-3$	$2.532875e+5$	$2.463930e-7$	1895.63
USL	$2.533441e+5$	$5.119095e+2$	$2.532923e+5$	$7.614912e+1$	2517.26
NERML	$2.581546e+5$	$2.171689e+4$	$2.540804e+5$	$3.754735e+3$	602.60

Table 6 Comparison of level methods for the SP instances

Inst.	NERML			APL			USL		
	Iter	gap	Time	Iter	gap	Time	Iter	gap	Time
SSN(50)	1,000	4.689592e − 1	497.45	90	4.379673e − 1	85.67	90	2.935656e − 1	177.16
SSN(100)	1,000	1.001421e + 0	1037.11	60	9.142773e − 1	114.92	60	8.150025e − 1	226.91
20-term(50)	1,000	1.058791e − 1	794.87	140	4.959911e − 2	301.40	590	4.277621e − 2	1632.86
20-term(100)	1,000	3.754735e + 3	1437.31	70	1.730272e + 3	197.45	110	1.740399e + 3	638.86

Table 7 Estimates on $\mathcal{D}_{v,Y}$

	SSN(50)	SSN(100)	20-term(50)	20-term(100)
Q	64	128	$1.68e+7$	$1.68e+7$

than those for the first two instances, see Table 7 for the estimates on $\mathcal{D}_{v,Y}$ reported by the USL method (with $Q_1 = 1$).

6 Concluding remarks

In this paper, we present new bundle-level type methods for convex programming. In particular, we show that both the ABL and APL methods are uniformly optimal for solving smooth, nonsmooth and weakly smooth problems without requiring the input of any smoothness information. We also demonstrate that, with little modification, the APL method is optimal for solving a class of composite CP problems. Based on the APL method, we develop a new smoothing technique, namely the USL method, which can achieve the optimal complexity for solving a class of saddle point problems without requiring the input of any problem parameters. We demonstrate the significant advantages of the APL and USL methods over some existing first-order methods for solving certain classes of semidefinite programing and stochastic programming problems.

Acknowledgments The author is very grateful to the co-editor Professor Adrian Lewis, the associate editor and two anonymous referees for their very useful suggestions for improving the quality and exposition of the paper.

7 Appendix

In this section, we provide the proof of Lemma 8.

Let F and F_η be defined in (4.11) and (4.13), respectively. Also let us denote, for any $\eta > 0$ and $x \in X$,

$$\psi_x(z) := F_\eta(x) + \langle \nabla F_\eta(x), z - x \rangle + \frac{\mathcal{L}_\eta}{2} \|z - x\|^2 + \eta \mathcal{D}_v, \tag{7.1}$$

where $\mathcal{D}_v$ and $\mathcal{L}_\eta$ are defined in (3.7) and (4.15), respectively. Clearly, in view of (1.2) and (4.16), ψ_x is a majorant of both F_η and f. Also let us define

$$Z_x := \left\{ z \in \mathbb{R}^n : \|z - x\|^2 = \frac{2}{\mathcal{L}_\eta} \left[\eta \mathcal{D}_v + F_\eta(x) - F(x) \right] \right\}. \tag{7.2}$$

Clearly, by the first relation in (4.16), we have

$$\|z - x\|^2 \le \frac{2\eta \mathcal{D}_v}{\mathcal{L}_\eta}, \quad \forall z \in Z_x. \tag{7.3}$$

Moreover, we can easily check that, for any $z \in Z_x$,

$$\psi_x(z) + \langle \nabla \psi_x(z), x - z \rangle = F(x), \tag{7.4}$$

where $\nabla \psi_x(z) = \nabla F_\eta(x) + \mathcal{L}_\eta (z - x)$.

The following results provides the characterization of a subgradient direction of F.

Lemma 9 *Let $x \in \mathbb{R}^n$ and $p \in \mathbb{R}^n$ be given. Then, $\exists z \in Z_x$ such that*

$$\langle F'(x), p \rangle \le \langle \nabla \psi_x(z), p \rangle = \langle \nabla F_\eta(x) + \mathcal{L}_\eta (z - x), p \rangle.$$

where $F'(x) \in \partial F(x)$.

Proof Let us denote

$$t = \frac{1}{\|p\|} \left\{ \frac{2}{\mathcal{L}_\eta} \left[\eta \mathcal{D}_v + F_\eta(x) - F(x) \right] \right\}^{\frac{1}{2}}$$

and $z_0 = x + tp$. Clearly, in view of (7.2), we have $z_0 \in Z_x$. By convexity of F and (7.4), we have

$$\begin{aligned} F(x) + \langle F'(x), tp\rangle \le F(x+tp) = \psi_x(z_0) &= F(x) + \langle \nabla\psi_x(z_0), z_0 - x\rangle \\ &= F(x) + t\langle \nabla\psi_x(z_0), p\rangle, \end{aligned}$$

which clearly implies the result. □

We are now ready to prove Lemma 8.

Proof of Lemma 8 First note that by the convexity of F, we have

$$F(x_0) - \big[F(x_1) + \langle F'(x_1), x_0 - x_1\big]\rangle \le \langle F'(x_0), x_0 - x_1\rangle + \langle F'(x_1), x_1 - x_0\rangle.$$

Moreover, by Lemma 9, $\exists z_0 \in Z_{x_0}$ and $z_1 \in Z_{x_1}$ s.t.

$$\begin{aligned} &\langle F'(x_0), x_0 - x_1\rangle + \langle F'(x_1), x_1 - x_0\rangle \\ &\quad\le \langle \nabla F_\eta(x_0) - \nabla F_\eta(x_1), x_0 - x_1\rangle + \mathcal{L}_\eta\langle z_0 - x_0 - (z_1 - x_1), x_0 - x_1\rangle \\ &\quad\le \mathcal{L}_\eta\|x_0 - x_1\|^2 + \mathcal{L}_\eta(\|z_0 - x_0\| + \|z_1 - x_1\|)\|x_0 - x_1\| \\ &\quad\le \mathcal{L}_\eta\|x_0 - x_1\|^2 + 2\mathcal{L}_\eta\left(\frac{2\eta\mathcal{D}_v}{\mathcal{L}_\eta}\right)^{\frac{1}{2}}\|x_0 - x_1\| \\ &\quad= \frac{\|A\|^2}{\sigma_v\eta}\|x_0 - x_1\|^2 + 2\left(\frac{2\|A\|^2\mathcal{D}_v}{\sigma_v}\right)^{\frac{1}{2}}\|x_0 - x_1\|, \end{aligned}$$

where the last inequality and equality follow from (7.3) and (4.15), respectively. Combining the above two relations, we have

$$\begin{aligned} F(x_0) - \big[F(x_1) + \langle F'(x_1), x_0 - x_1\rangle\big] &\le \frac{\|A\|^2}{\sigma_v\eta}\|x_0 - x_1\|^2 \\ &\quad + 2\left(\frac{2\|A\|^2\mathcal{D}_v}{\sigma_v}\right)^{\frac{1}{2}}\|x_0 - x_1\|. \end{aligned}$$

The result now follows by tending η to $+\infty$ in the above relation. □

References

1. Ahmed, S.: Smooth Minimization of Two-Stage Stochastic Linear Programs. Manuscript, Georgia Institute of Technology (2006)
2. Auslender, A., Teboulle, M.: Interior gradient and proximal methods for convex and conic optimization. SIAM J. Optim. **16**, 697–725 (2006)
3. Becker, S., Bobin, J., Candès, E.: NESTA: A Fast and Accurate First-Order Method for Sparse Recovery. Manuscript, California Institute of Technology (2009)
4. Ben-Tal, A., Nemirovski, A.S.: Lectures on Modern Convex Optimization: Analysis, Algorithms. Engineering Applications. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2000)

5. Ben-Tal, A., Nemirovski, A.S.: Non-euclidean restricted memory level method for large-scale convex optimization. Math. Program. **102**, 407–456 (2005)
6. Chenny, E.W., Goldstein, A.A.: Newton's methods for convex programming and tchebytcheff approximation. Numer. Math. **1**, 253–268 (1959)
7. d'Aspremont, A., Banerjee, O., El Ghaoui, L.: First-order methods for sparse covariance selection. SIAM J. Matrix Anal. Appl. **30**, 56–66 (2008)
8. Devolder, O., Glineur, F., Nesterov, Y.E.: First-Order Methods of Smooth Convex Optimization with Inexact Oracle. Manuscript, CORE, Université catholique de Louvain, Louvain-la-Neuve, Belgium (Dec 2010)
9. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. Technical report, SIAM J. Optim. (to appear) (2010)
10. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: a generic algorithmic framework. SIAM J. Optim. **22**, 1469–1492 (2012)
11. Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. SIAM J. Optim. **10**, 673–696 (2000)
12. Hoffman, A.: On approximate solutions of systems of linear inequalities. J. Res. Natl. Bureau Stand Sect. B Math. Sci. **49**, 263–265 (1952)
13. Juditsky, A., Nemirovski, A.S., Tauvel, C.: Solving Variational Inequalities with Stochastic Mirror-Prox Algorithm. Manuscript, Georgia Institute of Technology, Atlanta, GA (2008)
14. Kelley, J.E.: The cutting plane method for solving convex programs. J. SIAM **8**, 703–712 (1960)
15. Kiwiel, K.C.: An aggregate subgradient method for nonsmooth convex minimization. Math. Program. **27**, 320–341 (1983)
16. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Program. **46**, 105–122 (1990)
17. Kiwiel, K.C.: Proximal level bundle method for convex nondifferentable optimization, saddle point problems and variational inequalities. Math. Program. Ser. B **69**, 89–109 (1995)
18. Lan, G., Lu, Z., Monteiro, R.D.C.: Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming. Math. Program. **126**, 1–29 (2011)
19. Lan, G.: An optimal method for stochastic composite optimization. Math. Program. **133**(1), 365–397 (2012)
20. Lan, G., Nemirovski, A.S., Shapiro, A.: Validation analysis of mirror descent stochastic approximation method. Math. Program. **134**, 425–458 (2012)
21. Lemaréchal, C.: An extension of davidon methods to non-differentiable problems. Math. Program. Study **3**, 95–109 (1975)
22. Lemaréchal, C., Nemirovski, A.S., Nesterov, Y.E.: New variants of bundle methods. Math. Program. **69**, 111–148 (1995)
23. Lewis, A.S., Wright, S.J.: A Proximal Method for Composite Minimization. Manuscript, Cornell University, Ithaca, NY (2009)
24. Linderoth, J., Wright, S.: Decomposition algorithms for stochastic programming on a computational grid. Comput. Optim. Appl. **24**, 207–250 (2003)
25. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Ann. Oper. Res. **142**, 215–241 (2006)
26. Lu, Z.: Smooth optimization approach for sparse covariance selection. SIAM J. Optim. **19**, 1807–1827 (2009)
27. Mak, W.K., Morton, D.P., Wood, R.K.: Monte carlo bounding techniques for determining solution quality in stochastic programs. Oper. Res. Lett. **24**, 47–56 (1999)
28. Mosek. The Mosek Optimization Toolbox for Matlab Manual. Version 6.0 (revision 93). http://www.mosek.com
29. Nemirovski, A.S., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley-Interscience Series in Discrete Mathematics. Wiley, XV (1983)
30. Nemirovski, A.S.: Efficient Methods in Convex Programming. Lecture notes, Technion (1994)
31. Nemirovski, A.S.: Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM J. Optim. **15**, 229–251 (2005)
32. Nemirovski, A.S., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**, 1574–1609 (2009)

33. Nemirovskii, A.S., Nesterov, Y.E.: Optimal methods for smooth convex minimization. Zh. Vichisl. Mat. Fiz. **25**, 356–369 (1985). (In Russian)
34. Nesterov, Y.E.: Efficient Methods in Nonlinear Programming. Radio i Sviaz, Moscow (1989)
35. Nesterov, Y.E.: Gradient methods for minimizing composite objective functions. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, September (2007)
36. Nesterov, Y.E.: A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. Doklady AN SSSR **269**, 543–547 (1983)
37. Nesterov, Y.E.: On an approach to the construction of optimal methods of minimization of smooth convex functions. Ekonomo. i. Mat. Metody **24**, 509–517 (1988)
38. Nesterov, Y.E.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Massachusetts (2004)
39. Nesterov, Y.E.: Excessive gap technique in nonsmooth convex minimization. SIAM J. Optim. **16**, 235–249 (2005)
40. Nesterov, Y.E.: Smooth minimization of nonsmooth functions. Math. Program. **103**, 127–152 (2005)
41. Nesterov, Y.E.: Smoothing technique and its applications in semidefinite optimization. Math. Program. **110**, 245–259 (2007)
42. Oliveira, W., Sagastizábal, C., Scheimberg, S.: Inexact bundle methods for two-stage stochastic programming. SIAM J. Optim. **21**(2), 517–544 (2011)
43. Peña, J.: Nash equilibria computation via smoothing techniques. Optima **78**, 12–13 (2008)
44. Richtárik, P.: Approximate level method for nonsmooth convex minimization. J. Optim. Theory Appl. **152**(2), 334–350 (2012)
45. Ruszczyński, A.: Nonlinear Optimization, 1st edn. Princeton University Press, Princeton (2006)
46. Sagastizábal, C.: Nonsmooth Optimization: Thinking Outside of the Black Box. SIAG/OPT Views-and-News, pp. 1–9 (2011)
47. Sen, S., Doverspike, R.D., Cosares, S.: Network planning with random demand. Telecommun. Syst. **3**, 11–30 (1994)
48. Tseng, P.: On Accelerated Proximal Gradient Methods for Convex-Concave Optimization. Manuscript, University of Washington, Seattle (May 2008)