Modified Accelerated Bundle-Level Methods and Their Application in Two-Stage Stochastic Programming

Chunming Tang *©, Bo He and Zhenzhen Wang

College of Mathematics and Information Science, Guangxi University, Nanning 540004, China; bohe@st.gxu.edu.cn (B.H.); zzwang@mail.gxu.cn (Z.W.)
* Correspondence: cmtang@gxu.edu.cn

Received: 25 December 2019; Accepted: 11 February 2020; Published: 17 February 2020

Abstract: The accelerated prox-level (APL) and uniform smoothing level (USL) methods recently proposed by Lan (Math Program, 149: 1–45, 2015) can achieve uniformly optimal complexity when solving black-box convex programming (CP) and structure non-smooth CP problems. In this paper, we propose two modified accelerated bundle-level type methods, namely, the modified APL (MAPL) and modified USL (MUSL) methods. Compared with the original APL and USL methods, the MAPL and MUSL methods reduce the number of subproblems by one in each iteration, thereby improving the efficiency of the algorithms. Conclusions of optimal iteration complexity of the proposed algorithms are established. Furthermore, the modified methods are applied to the two-stage stochastic programming, and numerical experiments are implemented to illustrate the advantages of our methods in terms of efficiency and accuracy.

Keywords: stochastic programming; multi-step accelerated scheme; bundle method; complexity analysis

1. Introduction

In the fields of production planning, finance risk, telecommunication, and electricity, decision makers need to take into consideration uncertainty about the information and the model itself. Lack of data, calculation errors as well as unpredictability, etc. lead to the uncertainty in information. The uncertainty of model is derived from the structure of problem, the features of constraints, as well as the risks and profiles of decisions. Stochastic programming (SP) is an effective tool for dealing with optimization problems under uncertainty. The expectation model in stochastic programming is widely used, which maximizes the expectation of the benefit or minimizes the loss under the expected constraints. In this paper, we are concerned with the two-stage stochastic programming with recourse. Next, through a practical problem, network planning with random demand, we introduce the mathematical model of two-stage stochastic programming.

Due to demand for higher bandwidth and dedicated lines, network capacity is becoming a scarce resource. Consider a situation where a network provider plans bandwidth allocation between network links, under total network capacity \( b \) that is available for allocation. In addition, there are \( n \) different links needed to be expanded in capacity. The extra capacity allocated to link \( j \) is \( x_j \), where \( j = 1, 2, \ldots, n \) and vector \( x \in \mathbb{R}^n \) consists of elements \( x_j \). In network planning, demands refer to the number of connections requested between point-to-point pairs provided by the network at a certain time. Here, the demand \( D \in \mathbb{R}^m \) related to the \( m \) point-to-point pairs is modeled as a random variable.

Suppose that the extra capacities \( x_j, j = 1, 2, \ldots, n \) are given and the demand \( D \) is observed. Then the capacity planning model introduced in [1] is to minimize the total number of unoffered requests.
Let $i = 1, ..., m$ be the indices of point-to-point pairs, and the paths that can be offered to connections related to point-to-point pair $i$ be represented in $P(i)$. $c \in \mathbb{R}^n$ is a vector whose elements $c_i$ denote the capacity of current link $i$. $f_{ip}$ and $s_i$ denote the number of connections and unoffered requests related to pair $i$, respectively. For an observation $d = (d_1, ..., d_m)$ of the random variable $D$, one can obtain the optimal decision by solving the linear programming problem as follows

\[
\begin{align*}
\min & \quad \sum_{i=1}^m s_i \\
\text{s.t.} & \quad \sum_{i \in P(i)} A_{ip} f_{ip} \leq x + c, \\
& \quad \sum_{i \in P(i)} f_{ip} + s_i = d_i, \\
& \quad f_{ip} \geq 0, \quad s_i \geq 0, \quad i = 1, 2, ..., m, \quad \text{and} \quad p \in P(i).
\end{align*}
\] (1)

Here, 0-1 vector $A_{ip}$ is introduced as an incidence whose $j$-th component, $a_{ipj}$, is 1 if link $j$ lies in path $p$ and 0 otherwise. Let $Q(x, d)$ denote the optimal value of the above linear programming (1) which depends on the observation $d$ of $D$ and capacity $x$. It is obvious that $Q(x, D)$ is also a random variable due to the randomness of $D$. Thus, the capacities $x_j, j = 1, 2, ..., n$ can be obtained from the following programming problem

\[
\begin{align*}
\min & \quad \mathbb{E}[Q(x, D)] \\
\text{s.t.} & \quad \sum_{j=1}^n x_j \leq b, \\
& \quad x \geq 0.
\end{align*}
\] (2)

Here $\mathbb{E}[]$ refers to the expectation of the probability distribution of random variable $D$. In a word, the purpose of programming problem (2) is to minimize the expectation of unoffered requests, while satisfying the constraint within the total network capacity $b$. The above optimization model (1) and (2) is an instance of two-stage stochastic programming with recourse, in which (1) is the second-stage problem and (2) is the first-stage problem.

From the above practical instance we can see that the stochastic programming plays a key role in network planning with random demand. In addition to stochastic programming, the unit commitment problems in the field of energy systems usually involve mixed-integer quadratic programming. Recently, there have been some studies on solving such problems [2-4]. For solving the programming problems (1) to (2), Sen et al. [1] applied the stochastic decomposition method, which is one of the most efficient methods for stochastic programming. There are still many applications of stochastic programming in other fields, such as insurance and finance, valuation of electricity, telecommunications, hydrothermal power production planning and pollution control, etc. Stochastic models of these application problems can be found in [5]. In what follows, as an extension of the network planning problem, the standard mathematical model of two-stage stochastic programming with recourse is given, which is more convenient for generalization and theoretical analysis.

The two-stage stochastic programming with fixed recourse is in the following form

\[
\begin{align*}
\min & \quad c^T x + \mathbb{E}[Q(x, D)] \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0,
\end{align*}
\] (3)

where $x$ is the decision variable, $c$ is the the cost of production and $Q(x, D)$ denotes the optimal objective value of the second-stage problem

\[
\begin{align*}
\min & \quad q^T y \\
\text{s.t.} & \quad Wy = h - Tx, \\
& \quad y \geq 0.
\end{align*}
\] (4)

Here, $D$ is the demand vector with the elements $q^T, h^T, \text{and } T$. Let $W$ be the fixed recourse and $D = (D_1, ..., D_n)$ be a random variable with the support of probability distribution $\Xi \subset \mathbb{R}^d$. $\mathbb{E}[Q(x, D)]$ in the first-stage problem (3) is the expectation with respect to $D$. 

\[
= \int \mathbb{E}[Q(x, D)] \, d\Xi.
\]
The mathematical model of the two-stage stochastic programming is given and next we will summarize the methods mainly used to solve it. Since the two and multi-stage stochastic programming problems have very large dimension and special structures, they can be solved by means of decomposition. In two-stage models (3) and (4), suppose that the probability space $\Theta$ is finite, $\omega^s, s = 1, \ldots, S$ and that $p_s, s = 1, \ldots, S$ represent all basic events and their probabilities. Then the first-stage problem can be rewritten in the form

$$\begin{align*}
\text{min} & \quad c^T x + \sum_{s=1}^{S} p_s Q^s(x) \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0.
\end{align*}$$

(5)

Here, $Q^s(x)$ is the optimal value of the second-stage problem

$$\begin{align*}
\text{min} & \quad (q^s)^T y \\
\text{s.t.} & \quad W^s y = h^s - T^s x, \\
& \quad y \geq 0,
\end{align*}$$

(6)

where, $T^s = T(\omega^s), W^s = W(\omega^s), h^s = h(\omega^s), y^s = y(\omega^s), q^s = q(\omega^s)$.

The main idea of the basic dual decomposition methods is to establish certain approximations to the first-stage problem by means of solving subproblems in structure (6). As an original decomposition-type method, the cutting-plane method establishes a linear model of $\sum_{s=1}^{S} p_s Q^s(x)$, and its main scheme can be found in [6,7]. Simplicity and cheapness in calculation are advantages of the cutting-plane model. With the increasing of cuts, however, the demand for storage grows which is a typical difficulty of such method. On the other hand, although there is a good initial iteration, such method may generate a larger stepsize. In order to tackle these difficulties, Ruszczynski [8] proposed the regularized decomposition method which is considered as an improvement of the basic one. Another way to avoid generating longer steps is the trust region method which was extended to two-stage stochastic programming by Linderoth and Wright [9]. In addition, bundle methods, which can be viewed as stabilized variants of the original cutting-plane methods, were developed in [10–12]. Algorithmic modifications based on bundle methods were introduced in [13,14]. The bundle level (BL) method was first proposed by Lemaréchal et al. [15] as another kind of bundle method. On this basis, in [16–18] the “restricted-memory” version of BL method is developed which performs well in numerical experiments. In recent years, there has been tremendous development in “asynchronous” and “partial” versions of BL methods, see references [19–23]. Considering that the research on BL methods is only focused on general non-smooth convex optimization (CP) problems, Lan [24] proposed an accelerated BL-type method, namely the accelerated bundle level (ABL) method and its restricted memory version, the accelerated prox-level (APL) method. Benefiting from the multi-step strategy introduced by Nesterov [25] and later applied in [26–30], both ABL and APL methods are uniformly optimal for solving non-smooth, weakly smooth and smooth CP problems. In addition, by incorporating Nesterov’s smoothing technique [16,17] into APL method, Lan [24] presented the uniform smoothing level (USL) method for solving structure non-smooth CP problems with optimal iteration complexity. In particular, the USL method does not require any input of problem parameters. Moreover, Lan [24] illustrated that the APL and USL methods, when applied to solving semidefinite programming and two-stage stochastic programming, have obvious advantages in computation time and accuracy over the related gradient-type algorithms and some existing methods.

Our main work in this paper includes several aspects. First of all, on the basis of Lan’s work, we make further improvement and present the modified accelerated prox-level (MAPL) method. By selecting the proper proximal function, the MAPL method only need to solve one subproblem to update the prox-center and the lower bound synchronously, which improves the computational efficiency. In addition, MAPL method can achieve uniformly optimal complexity for solving smooth, weakly smooth, and non-smooth CP problems uniformly. Furthermore, we extend the MAPL
method to solve the structure non-smooth CP problems and present the modified uniform smoothing level (MUSL) method. Finally, we apply the proposed methods to solve the two-stage stochastic programming problems with recourse. The numerical results show that the MAPL and MUSL methods have certain advantages in iterations and computation time.

The present paper is built up as follows. In Section 2, some related works about BL-type methods are reviewed. The MAPL method and its complexity analysis are presented in Section 3. The MUSL method and its complexity analysis are presented in Section 4. The application of MAPL and MUSL methods to two-stage stochastic programming as well as numerical experiments are shown in Section 5. Conclusions are presented in the final section.

2. Related Work

This section reviews some related work on BL methods. Specifically, the main ideas of BL methods and related notation are reviewed in Section 2.1. The APL method and its gap reduction procedure are introduced in Section 2.2. Finally, we apply the proposed methods to solve the two-stage stochastic CP problems and present the modified uniform smoothing level (MUSL) method.}

2.1. The Bundle Level (BL) Method

Consider the general CP problem

\[ f^* := \min_{x \in X} f(x), \]  

where the constraint set \( X \subseteq \mathbb{R}^n \) is convex and compact, and the objective function \( f : X \to \mathbb{R} \) is closed and convex over \( X \). The function \( f \) is known only through a first-order oracle returning the function value \( f(x) \) and a subgradient \( f'(x) \in \partial f(x) \) for a given point \( x \in X \), where \( \partial f(x) := \{ \xi \in \mathbb{R}^n \mid f(y) \geq f(x) + \langle \xi, y - x \rangle, \forall y \in \mathbb{R}^n \} \) is the subdifferential of \( f \) at \( x \).

Given a sequence of iteration points, the first-order information \( \{ f(x_k) \} \) and \( \{ f'(x_k) \} \) are provided by oracle. The cutting-plane approximation of \( f \) is generated by

\[ \tilde{f}_k(x) := \max\{ h(x_i, x) : 1 \leq i \leq k \} \leq f(x), \]

where \( h(x_i, x) := f(x_i) + \langle f'(x_i), x - x_i \rangle \). In addition, \( \langle x, y \rangle := \sum_i x_i y_i, x, y \in \mathbb{R}^n \) refers to the inner product. The level set \( X_k \) of \( \tilde{f}_k \) with level parameter \( \ell_k \) is defined by \( X_k := \{ x \in X : f_k(x) \leq \ell_k \} \). The BL method [15] generates the next iteration point \( x_{k+1} \) by

\[ x_{k+1} = \text{argmin}_{x \in X_k} \| x - x_k \|^2. \]  

(8)

The main steps of the BL method are listed below.

**Step 1.** Set \( f_k := \min \{ f(x_i) \mid 1 \leq i \leq k \} \) and compute a lower bound \( \tilde{f}_k := \min_{x \in X} f_k(x) \);

**Step 2.** Set the level parameter \( \ell_k = \beta f_k + (1 - \beta) \tilde{f}_k \) for some \( \beta \in (0, 1) \);

**Step 3.** Set \( X_k := \{ x \in X : f_k(x) \leq \ell_k \} \) and generate the new iterate by solving (8).

2.2. The Accelerated Prox-Level (APL) Method and Its Gap Reduction Procedure \( G_{MAPL} \)

In this subsection we consider the CP problem (7), where \( f \) satisfies the following inequality

\[ f(y) - f(x) - \langle f'(x), y - x \rangle \leq \frac{M}{1 + \rho} \| y - x \|^1 + \rho, \forall x, y \in X, \]

(9)

for some \( M > 0 \), \( \rho \in [0, 1] \) and \( f'(x) \in \partial f(x) \). It is easy to show that non-smooth (\( \rho = 0 \)), smooth (\( \rho = 1 \)) and weakly smooth (\( \rho \in (0, 1) \)) CP problems are contained in this family of problems. Lan [24] generalized the BL method to accelerated bundle level (ABL) and accelerated prox-level (APL) methods, such that they achieve uniformly optimal complexity bounds for functions satisfying (9). Here, we mainly introduce the APL method and its principle.
Firstly, Lan [24] introduced three related iteration sequences, \( \{x_k^1\} \), \( \{x_k^n\} \) and \( x_k \) to establish the cutting-plane approximation \( \bar{f}_k(x) \), generate the upper bounds \( \bar{f}_k \), and update prox-center \( x_k \), respectively. Specifically, \( x_k^1 \) and \( x_k^n \) are updated by \( x_k^1 = (1 - \alpha_k)x_k^n + \alpha_k x_{k-1} \) and \( x_k^n = (1 - \alpha_k)x_{k-1}^n + \alpha_k x_k \). Secondly, Lan introduced an internal procedure \( G_{APL} \) to reduce the gap between the upper and lower bounds of \( f^* \). The algorithmic framework of the APL method is as follows (Algorithm 1).

**Algorithm 1:** The APL method.

**Input:** Choose \( p_0 \in X \), stopping tolerance \( \epsilon > 0 \) and parameters \( \lambda, \theta \in (0, 1) \).

**Step 0:** (Initialization) Set \( p_1 \in \text{Argmin}_{x \in X} h(p_0, x) \), \( lb_1 = h(p_0, p_1) \) and \( ub_1 = f(p_1) \).

Let \( s = 1 \).

**Step 1:** (Stopping test) If \( ub_s - lb_s \leq \epsilon \), terminate.

**Step 2:** (Call \( G_{APL} \) procedure) Set \( (p_{s+1}, lb_{s+1}) = G_{APL}(p_s, lb_s, \lambda, \theta) \) and \( ub_{s+1} = f(p_{s+1}) \).

**Step 3:** (Loop) Set \( s = s + 1 \), and return to Step 1.

Next, we focus on describing the gap reduction procedure \( G_{APL} \). Denote the level set \( X_f(\ell) := \{ x \in X : f(x) \leq \ell \} \). For a given iterate \( z \), denote

\[
\bar{f} := \min \{ h(z, x) : x \in X_f(\ell) \}. \tag{10}
\]

Then, it can be verified that

\[
\min \{ \ell, \bar{f} \} \leq f(x), \quad \forall x \in X.
\]

This implies that \( \min \{ \ell, \bar{f} \} \) a lower bound of \( f^* \). However, the problem (10) is difficult to solve in general. To overcome the difficulty and obtain the lower bound in a convenient way, Lan used a compact and convex set \( X' \) to replace \( X_f(\ell) \) in problem (10). Thus, one can solve the relaxation of (10)

\[
\bar{f} := \min \{ h(z, x) : x \in X' \}.
\]

Here, the set \( X' \) is called the localizer of the level set \( X_f(\ell) \) satisfying \( X_f(\ell) \subseteq X' \subseteq X \). Then we obtain a lower bound on \( f^* \) as follow

\[
\min \{ \ell, \bar{f} \} \leq f(x), \quad \forall x \in X. \tag{11}
\]

Indeed, as shown in [24, Lemma 4], if \( X' \neq \emptyset \), then \( \ell \leq \min \{ h(z, x) : x \in X_f(\ell) \} \leq f(x) \), for all \( x \in X \). If \( X' = \emptyset \), we have that \( \bar{f} = +\infty \) and \( X_f(\ell) = \emptyset \), and therefore \( f(x) \geq \ell \) for all \( x \in X \). Thus, (11) holds.

Moreover, in order to make better use of the structure of the feasible set \( X \), similar to the NERML algorithm in [16,17], Lan introduced the prox-function to replace the Euclidean distance function \( \| \cdot \|^2 \). Here, a function \( \omega : X \to \mathbb{R} \) is served as a prox-function of a convex compact set \( X \subseteq \mathbb{R}^n \) with coefficient \( \sigma_\omega \), if it is a function with differentiability, strong convexity as well as coefficient \( \sigma_\omega \), i.e.

\[
\langle \nabla \omega(x) - \nabla \omega(z), x - z \rangle \geq \sigma_\omega \| x - z \|^2, \quad \forall x, z \in X. \tag{12}
\]

Furthermore, one can redefine the diameter of \( X \) with \( d_\omega \) by

\[
D^2_{\omega,X} := \max \{ \omega(x) - \omega(z) - \langle \nabla \omega(z), x - z \rangle, \quad x, z \in X \}. \tag{13}
\]

This lead to the following relation

\[
\| x - z \|^2 \leq (2/\sigma_\omega)D^2_{\omega,X} =: \Omega_{\omega,X}, \quad \forall x, z \in X. \tag{14}
\]
Furthermore, let
\[ d_\omega(x) := \omega(x) - [\omega(x_0) + \langle \omega'(x_0), x - x_0 \rangle], \forall x \in X, \] (15)
be the prox-function on \( X \) and \( x_0 := \arg \min_{x \in X} d_\omega(x) \) be the prox-center of \( d_\omega(x) \). It follows that
\[ d_\omega(x) \geq (1/2)\sigma_\omega \| x - x_0 \|^2, \] (16)
and \( d_\omega(x) \) is a strong convex function with coefficient \( \sigma_\omega \).

The internal gap reduction procedure \( G_{\text{APL}} \) of APL method is as follows.

**The APL gap reduction procedure:** \( (p^+, \text{lb}^+) = G_{\text{APL}}(p, \text{lb}, \lambda, \theta) \)

**Step 0: (Initialization)** Set \( x_0^u = p, \overline{f}_0 = f(x_0^u), f_0 = \text{lb} \) and \( \ell = \lambda f_0 + (1 - \lambda) \overline{f}_0 \). In addition, choose \( x_0 \in X \) and the initial localizer \( X_0' = X \). The prox-function \( d_\omega(x) \) is defined in (15). Also let \( k = 1 \).

**Step 1: (Update lower bound)** Let \( x_k^u = a_k x_{k-1} + (1 - a_k) x_{k-1}^u \) and \( h(x_k^u, x) = f(x_k^u) + \langle f'(x_k^u), x - x_k^u \rangle \).
\[ \text{lb}_k := \min_{x \in X_{k-1}} \left\{ h(x_k^u, x) \right\}, \text{ and } f_k := \max \left\{ f_{k-1}, \min \{ \ell, \text{lb}_k \} \right\}. \]
If \( f_k \geq \ell - \theta(\ell - f_0) \), then **stop** the procedure and output \( p^+ = x_k^u \) and \( \text{lb}^+ = f_k \).

**Step 2: (Update prox-center)** Set
\[ x_k := \arg \min_{x \in X_{k-1}} \left\{ d_\omega(x) : h(x_k^u, x) \leq \ell \right\}. \]

**Step 3: (Update upper bound)** Set \( \overline{f}_k = \min \{ \overline{f}_{k-1}, f(a_k x_k + (1 - a_k) x_{k-1}^u) \} \) and \( x_k^u \) such that \( f(x_k^u) = \overline{f}_k \). If \( \overline{f}_k \leq \ell + \theta(\overline{f}_0 - \ell) \), then **stop** the procedure and output \( p^+ = x_k^u \) and \( \text{lb}^+ = f_k \).

**Step 4: (Update localizer)** Choose \( X_k' \) such that \( X_k \subseteq X_k' \subseteq X_k \), where
\[ X_k := \left\{ x \in X_{k-1} : h(x_k^u, x) \leq \ell \right\} \text{ and } X_k := \left\{ x \in X : \langle \nabla d_\omega(x_k), x - x_k \rangle \geq 0 \right\}. \]

**Step 5: (Loop)** Set \( k = k + 1 \) and return to Step 1.

3. The Modified Accelerated Prox-Level (MAPL) Method

In this section, we proposed the modified accelerated prox-level (MAPL) method which requires only one subproblem to be solved per iteration while achieving the uniformly optimal iteration complexity for solving the black-box CP problems. We first present the modified gap reduction procedure \( G_{\text{MAPL}} \) which generates a new search point \( p^+ \) and a new lower bound \( \text{lb}^+ \) of \( f^k \) such that \( f(p^+) - \text{lb}^+ \leq q[f(p) - \text{lb}] \) from the given input \( p \) and \( \text{lb} \) and some \( q \in (0, 1) \). Here, the value of \( q \) depends on the parameters \( \lambda, \theta \in (0, 1) \).

**The MAPL gap reduction procedure:** \( (p^+, \text{lb}^+) = G_{\text{MAPL}}(p, \text{lb}, \lambda, \theta) \)

**Step 0: (Initialization)** Set \( x_0^u = p, \overline{f}_0 = f(x_0^u), f_0 = \text{lb} \) and \( \ell = \lambda f_0 + (1 - \lambda) \overline{f}_0 \). In addition, choose \( x_0 \in X \) and the initial localizer \( X_0' = X \). The prox-function \( d_\omega(x) \) is defined in (15). Let \( k = 1 \).

**Step 1: (Update level set)** Set
\[ x_k^u = a_k x_{k-1} + (1 - a_k) x_{k-1}^u, \] (17)
\[ h(x_k^u, x) = f(x_k^u) + \langle f'(x_k^u), x - x_k^u \rangle, \] (18)
\[ X_k := \left\{ x \in X_{k-1} : h(x_k^u, x) \leq \ell \right\}. \] (19)

**Step 2: (Update prox-center and lower bound)** Set
\[ x_k = \arg \min_{x \in X_k} d_\omega(x). \] (20)
If $X_k = \emptyset$, then stop the procedure and output $p^+ = x_{k-1}^u, \text{lb}^+ = \ell$.

**Step 3: (Update upper bound)**

Set

\[ x_{k+1}^u = \alpha_k x_k + (1 - \alpha) x_{k-1}^u, \]

\[ x_k^u = \begin{cases} x_k^{u'}, & \text{if } f(x_k^{u'}) < \tilde{f}_{k-1}, \\ x_{k-1}^u, & \text{otherwise}, \end{cases} \]

and $\tilde{f}_k = f(x_k^u)$. If $\tilde{f}_k \leq \ell + \theta(f_0 - \ell)$, then stop the procedure (there is a significant improvement on the upper bound) and output $p^+ = x_0^u, \text{lb}^+ = \text{lb}$.

**Step 4: (Update localizer)**

Choose an arbitrary $X_0' \subset X_0 \subset X_k$, where

\[ X_k = \{ x \in X : \langle \nabla d_\omega(x_k), x - x_k \rangle \geq 0 \}. \]

**Step 5: (Loop)**

Set $k = k + 1$ and return to Step 1.

The following are a few remarks on the MAPL gap reduction procedure. Firstly, the upper bound $\tilde{f}_0 = f(x_0^u)$ and lower bound $f_0 = \text{lb}$ of $f^*$ in Step 0 are obtained from the outer iteration of the MAPL method (described below), and $\tilde{f}_0$ and $f_0$ are fixed throughout the entire process of $\mathcal{G}_{\text{MAPL}}$. Furthermore, the level parameter $\ell$ is also fixed as a convex combination of $\tilde{f}_0$ and $f_0$. But in the original BL methods, the level parameter changes in each iteration. Secondly, Step 2 and Step 3 have two exits. When the procedure stops at Step 2, $\ell$ is the lower bound on $f^*$; when the procedure stops at Step 3, there is a significant progress on $\tilde{f}_k$, which depends on the parameter $\theta$. Compared with the gap reduction procedure $\mathcal{G}_{\text{APL}}$ of APL method, the $\mathcal{G}_{\text{MAPL}}$ updates the lower bound in a easier way. Indeed, in Step 1 of $\mathcal{G}_{\text{APL}}$ see [24], in order to determine whether the lower bound $\text{lb}$ needs to be updated, a linear programming subproblem is first solved. However, in the $\mathcal{G}_{\text{MAPL}}$ procedure, the prox-center $x_k$ and the lower bound $\text{lb}$ are merged into one subproblem to be updated by selecting appropriate prox-function $d_\omega$. While solving the subproblem (20), it can be automatically checked that whether $X_k = \emptyset$. If $X_k = \emptyset$, the lower bound $\text{lb}$ is directly updated to $\ell$, which avoids solving the linear programming as $\mathcal{G}_{\text{APL}}$, and reduce the number of calculation. Thirdly, in Step 4, $X_k'$ can be selected arbitrarily under condition $X_k \subset X_k' \subset X_k$. However, the number of constraints in $X_k'$ increases with $k$, and $X_k$ has only one more constraint than the feasible set $X$. In practice, we can choose $X_k'$ between these two sets, which can control the number of constraints in (19) and reduce the computation cost. Moreover, a proper selection of the stepsize sequences $\{\alpha_k\}$ is critical for $\mathcal{G}_{\text{MAPL}}$ to terminate after finite iterations and achieve optimal iteration complexity. Lan proposed a generalized selection rule for the sequence $\{\alpha_k\}$ in [24]. Chen et al. [31] proposed a more concise selection rule, i.e., the sequence $\{\alpha_k\}$ satisfies the following conditions:

\[ a_1 = 1, 0 < a_k \leq 1, a_k \leq \frac{c}{k}, \frac{1 - a_{k+1}}{a_k} \leq \frac{1}{\ell^{1+\gamma}}, \forall k \geq 1, \]

for any $\gamma \in [0, 1]$ and some $c > 0$. Two examples for $\{\alpha_k\}$ are as follows [31]:

1. $a_k = 2/(k + 1), k = 1, 2, \ldots, \text{with } c = 2;$
2. $a_1 = 1, a_{k+1} = (1 - a_{k+1})a_k^{2}, \text{with } c = 2.$

The following lemma shows the important properties of procedure $\mathcal{G}_{\text{MAPL}}$. These properties are similar to those of [24,31], whose proof can be found in Appendix A.

**Lemma 1.** The following properties hold for the procedure $\mathcal{G}_{\text{MAPL}}$.

a. \{ $X_k'$ \}_{k \geq 0} is a collection of localizers for the level set $X_f(\ell)$ ;

b. $\tilde{f}_0 \geq \tilde{f}_1 \geq \cdots \geq \tilde{f}_k \geq f^*$ holds for any $k \geq 1$;

c. $X_f(\ell) \neq \emptyset$, and hence $X_f(\ell) \subseteq X_k \subseteq X_k' \subseteq X_k$ for any $k \geq 1$;
We have for $x$.

**Proof.** Assume that the gap reduction procedure $\Delta$ where $d\mathbb{G}$ the prox-function, then the number of iterations of the procedure $\mathbb{G}_\text{MAPL}$ stops, the relation $f(p^+) - \text{lb}^+ \leq q(f(p) - \text{lb})$ holds, where

$$q \equiv q(\lambda, \theta) = \max \{ \lambda, 1 - \lambda(1 - \theta) \}. \quad (25)$$

Referring to the theoretical analysis mode of [24], the following proposition will show that the gap between the upper bound $f(x^u_k)$ and the level parameter $\ell$ will decrease with $k$, and it is proven that when the algorithm stops, the total number of iterations does not exceed an upper bound.

**Proposition 1.** If $\alpha_k$ satisfy the condition (24), take $\gamma = \rho$, let $\ell$ be the level parameter and $d_\omega(x)$ denotes the prox-function, then the number of iterations of the procedure $\mathbb{G}_\text{MAPL}$ does not exceed

$$N(\Delta_0) := \left[ \left( \frac{e^{1+\rho}M\Omega^{1/\rho}_{\omega, X}}{\lambda(1 + \rho)\Delta_0} \right)^{\frac{2}{1+\rho}} \right]. \quad (26)$$

where $\Delta_0 = f_0 - f_0$ and $\Omega_{\omega, X}$ is defined in (14).

**Proof.** Assume that the gap reduction procedure $\mathbb{G}_\text{MAPL}$ does not stop at the $K$-th iteration. From (16) we have for $x_1 \in X$

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

By the definition of $X_k$ in (23) we have $\langle \nabla d_\omega(x_k), z - x_k \rangle \geq 0$, for any $z \in X_k$. This together with $X'_k \subseteq X_k$ shows that

$$\langle \nabla d_\omega(x_k), z - x_k \rangle \geq 0, \forall z \in X'_k.$$

It follows from (19) and (20) that $x_{k+1} \in X_{k+1} \subseteq X'_k$, for all $1 \leq k \leq K$. Moreover, due to the strong convexity of $d_\omega(x)$ and the optimality condition of subproblem (20), we have

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

It turns out that

$$\frac{\sigma_\omega}{2} \|x_k - x_{k-1}\|^2 \leq d_\omega(x_{k+1}) - d_\omega(x_k), \forall k \geq 1.$$ 

Summing up the above inequalities over $k$, we obtain that

$$\frac{\sigma_\omega}{2} \sum_{k=1}^{K} \|x_{k+1} - x_k\|^2 \leq d_\omega(x_K). \quad (27)$$

According to (22), (9) and (18), we have

$$f(x^u_k) \leq f(x^u_k) \leq f(x^u_k) + \langle f'(x^u_k), x^u_k - x^L_k \rangle + \frac{M}{1 + \rho} \|x^u_k - x^L_k\|^{1+\rho}$$

$$= h(x^u_k, x^L_k) + \frac{M}{1 + \rho} \|x^u_k - x^L_k\|^{1+\rho}.$$
From (18), (21), the convexity of \( f \), (19) and (22), we have

\[
h(x_{k}^l, x_{k}^u) = f(x_{k}^l) + (f'(x_{k}^l), (1 - a_k)x_{k-1}^u + a_k x_k - x_{k}^l) \\
= (1 - a_k)[f(x_{k}^l) + (f'(x_{k}^l), x_{k-1}^u - x_{k}^l)] + a_k [f(x_{k}^l) + (f'(x_{k}^l), x_k - x_{k}^l)] \\
= (1 - a_k)h(x_{k-1}^u, x_k) + a_k h(x_{k}^l, x_k) \\
\leq (1 - a_k)f(x_{k-1}^u) + a_k h(x_{k}^l, x_k) \\
\leq (1 - a_k)f(x_{k-1}^u) + a_k \ell.
\]

By the definition of \( \tilde{x}_k^u \) (21) and \( x_k^l \) (17), it is easy to show that

\[
\tilde{x}_k^u - x_k^l = a_kx_k + (1 - a_k)x_{k-1}^u - a_kx_{k-1} - (1 - a_k)x_{k-1}^u \\
= a_kx_k - a_kx_{k-1} = a_k(x_k - x_{k-1}).
\]

It then follows that

\[
f(x_{k}^u) \leq (1 - a_k)f(x_{k-1}^u) + a_k \ell + \frac{Ma_k^{1+\rho}}{1+\rho} \|x_k - x_{k-1}\|^{1+\rho}.
\]

Subtracting both sides of the above inequality by \( \ell \) and dividing both sides by \( a_k^{1+\rho} \), from (24), we obtain that

\[
\frac{1}{a_k^{1+\rho}}[f(x_{k}^u) - \ell] \leq \frac{1}{a_k^{1+\rho}}[f(x_{k-1}^u) - \ell] + \frac{M}{1+\rho} \|x_k - x_{k-1}\|^{1+\rho} \\
\leq \frac{1}{a_{k-1}^{1+\rho}}[f(x_{k-1}^u) - \ell] + \frac{M}{1+\rho} \|x_k - x_{k-1}\|^{1+\rho}, \quad \forall k \geq 2.
\]

As a result,

\[
\frac{1}{a_1^{1+\rho}}[f(x_{1}^u) - \ell] \leq \frac{M}{1+\rho} \|x_1 - x_0\|^{1+\rho},
\]

\[
\frac{1}{a_2^{1+\rho}}[f(x_{2}^u) - \ell] \leq \frac{1}{a_1^{1+\rho}}[f(x_{1}^u) - \ell] + \frac{M}{1+\rho} \|x_2 - x_1\|^{1+\rho},
\]

\[
\vdots
\]

\[
\frac{1}{a_K^{1+\rho}}[f(x_{K}^u) - \ell] \leq \frac{1}{a_{K-1}^{1+\rho}}[f(x_{K-1}^u) - \ell] + \frac{M}{1+\rho} \|x_K - x_{K-1}\|^{1+\rho}.
\]

Summing up the above inequality over \( m \) and from the fact that \( a_k > 0, \forall k \geq 1 \), we have

\[
f(x_{K}^u) - \ell \leq \frac{a_1^{1+\rho}M}{1+\rho} \sum_{k=1}^{K} \|x_k - x_{k-1}\|^{1+\rho}.
\]

(28)

Applying the Hölder inequality to the above inequality and using (27), we have

\[
\sum_{k=1}^{K} \|x_k - x_{k-1}\|^{1+\rho} \leq K^{\frac{1}{1+\rho}} \left( \sum_{k=1}^{K} \|x_k - x_{k-1}\|^2 \right)^{\frac{1+\rho}{2}} \\
\leq K^{\frac{1}{1+\rho}} \left( \frac{2d_{\omega} (x_K)}{c_{\omega}} \right)^{\frac{1+\rho}{2}} \leq K^{\frac{1}{1+\rho}} \left( \frac{2D_{\omega}}{c_{\omega}} \right)^{\frac{1+\rho}{2}}.
\]
Applying (24) and (14) to (28), as well as the above inequality, we have
\[ f(x_u^K) - \ell \leq \frac{c^{1+\rho}M}{K^{1+\rho}(1+\rho)} K^{1-\frac{1}{\rho}} \left( \frac{2D_{\omega,\lambda}}{\sigma_{\omega}} \right)^{\frac{1}{\rho}} \leq \frac{c^{1+\rho}M}{1+\rho} \Omega_{\omega,\lambda}K^{\frac{1}{1-\frac{1}{\rho}}}. \] (29)

From the fact that \( G_{MAPL} \) does not stop at Step 3, we have
\[ f_k - \ell = f(x_u^K) - \ell > \theta(\bar{f}_0 - \ell) = \theta \lambda(\bar{f}_0 - f_0) = \theta \lambda(\ub - \lb) = \lambda \Delta_0. \]
This together with (29) shows that
\[ \lambda \Delta_0 \leq \frac{M c^{1+\rho} K^{\frac{1}{1-\frac{1}{\rho}}}}{1+\rho} \Omega_{\omega,\lambda} + \rho_0 \omega, X K^{\frac{1}{1-\frac{1}{\rho}}}. \]

We finally conclude that
\[ K < \left( \frac{c^{1+\rho} M \Omega_{\omega,\lambda} K^{\frac{1}{1-\frac{1}{\rho}}}}{\lambda \theta (1+\rho) \Delta_0} \right)^{\frac{2}{1-\frac{1}{\rho}}}, \]
that is
\[ N(\Delta_0) = \left[ \left( \frac{c^{1+\rho} M \Omega_{\omega,\lambda} K^{\frac{1}{1-\frac{1}{\rho}}}}{\lambda \theta (1+\rho) \Delta_0} \right)^{\frac{2}{1-\frac{1}{\rho}}} \right]. \]

\[ \Box \]

After giving the relevant properties and complexity analysis of the procedure \( G_{MAPL} \), we are now in a position to introduce the so-called modified accelerated prox-level (MAPL) method, which repeatedly calls the procedure \( G_{MAPL} \) during iteration progress until it finds an approximate solution with the given accuracy. The algorithmic framework of MAPL method is as follows (Algorithm 2).

**Algorithm 2: The MAPL method.**

**Parameters:** Choose stopping tolerance \( \varepsilon > 0 \) and parameters \( \lambda, \theta \in (0,1) \).

**Step 0: (Initialization)** Choose initial point \( p_0 \), set \( p_1 \in \arg \min_{x \in X} h(p_0, x) \), \( \lb_1 = h(p_0, p_1) \) and \( \ub_1 = \min\{f(p_0), f(p_1)\} \). Let \( s = 1 \);

**Step 1: (Stopping test)** If \( \ub_s - \lb_s \leq \varepsilon \), terminate;

**Step 2: (Call \( G_{MAPL} \) procedure)** Set \( (p_{s+1}, \lb_{s+1}) = G_{MAPL}(p_s, \lb_s, \lambda, \theta) \) and \( \ub_{s+1} = f(p_{s+1}) \);

**Step 3: (Loop)** Set \( s = s + 1 \), and return to Step 1.

Since the procedure \( G_{MAPL} \) is called during the progress of the MAPL method, we consider that an iteration of \( G_{MAPL} \) is also an iteration of the MAPL method. Take this fact into consideration, the following theorem establishes the convergence and iteration complexity of the MAPL method.

**Theorem 1.** For given \( \varepsilon > 0 \), if \( a_k \in (0,1], k = 1, 2, \ldots \), in \( G_{MAPL} \) are chosen to satisfy condition (24) with some \( c > 0 \), then

1. The number of procedure \( G_{MAPL} \) called by the MAPL method can be bounded by
\[ S(\varepsilon) := \max \left\{ 0, \log_{\frac{1}{2}} \left( \frac{M \Omega_{\omega,\lambda}}{1+\rho) \varepsilon} \right) \right\}; \]
(2) The total number of iterations performed by the MAPL method does not exceed
\[
S(\epsilon) + \frac{1}{1 - q^{-2}} \left( \epsilon^{1+\rho} \frac{M \Omega_{u,X}^{1+\rho}}{(1 + \rho)\lambda \theta \epsilon} \right)^{\frac{2}{1+\rho}}.
\]

**Proof.** (1) Without loss of generality, we suppose that \( \Delta_1 > \epsilon \), where \( \Delta_s = ub_s - lb_s, s \geq 1 \). According to Step 1 of the MAPL method, (9) and (14), we have
\[
\Delta_1 = f(p_1) - h(p_0, p_1) = f(p_1) - [f(p_0) + \langle f'(p_0), p_1 - p_0 \rangle] \leq \frac{M\|p_1 - p_0\|^{1+\rho}}{1 + \rho} \leq \frac{M \Omega_{u,X}^{1+\rho}}{1 + \rho}.
\]

From Lemma 1 and the fact that \( ub_s = f(p_s) \), we have
\[
\Delta_{s+1} \leq q \Delta_s, \ s \geq 1.
\]

Furthermore, suppose that the MAPL method finds an \( \epsilon \)-solution after calling \( S(\epsilon) \) times \( G_{MAPL} \), i.e., the MAPL method stops if \( s := S(\epsilon) + 1 \). Then we have
\[
\Delta_{S(\epsilon)+1} \leq \epsilon \leq \Delta_{S(\epsilon)},
\]
and
\[
\epsilon < q^{S(\epsilon)-1} \Delta_1 \leq q^{S(\epsilon)-1} M \Omega_{u,X}^{1+\rho}.
\]
It is easy to obtain
\[
\left( \frac{1}{q} \right)^{S(\epsilon)-1} \leq \frac{M \Omega_{u,X}^{1+\rho}}{(1 + \rho)\epsilon},
\]
that is
\[
S(\epsilon) := \left[ \max \left\{ 0, \log_{\frac{1}{q}} \left( \frac{M \Omega_{u,X}^{1+\rho}}{(1 + \rho)\epsilon} \right) \right\} \right].
\]

(2) Suppose that the procedure \( G_{MAPL} \) has been called \( S(\epsilon) \) times in MAPL method. Then by (30) and (31) we have
\[
\Delta_s > \epsilon q^{s-S(\epsilon)}, s = 1, 2, ..., S(\epsilon).
\]
Due to \( \Delta_{S(\epsilon)} > \epsilon \), we know that
\[
\sum_{s=1}^{S(\epsilon)} \Delta_s \leq \sum_{s=1}^{S(\epsilon)} q^{\frac{2}{1+\rho}(s-S(\epsilon))} = \sum_{t=0}^{S(\epsilon)-1} q^{\frac{2}{1+\rho}} \leq \frac{1}{(1 - q^{-2})} \frac{1}{\epsilon^{\frac{2}{1+\rho}}}.
\]
This together with Lemma 1 and Proposition 1 shows that the total number of iterations performed by the MAPL method does not exceed
\[
S(\epsilon) + \frac{1}{1 - q^{-2}} \left( \epsilon^{1+\rho} \frac{M \Omega_{u,X}^{1+\rho}}{(1 + \rho)\lambda \theta \epsilon} \right)^{\frac{2}{1+\rho}}.
\]

Here, we denote the number of the internal iterations performed by the s-th procedure \( G_{MAPL} \) with \( N_s \). \( \Box \)

We present a few remarks on the iteration complexity of MAPL method.
Remark 1. According to the classic complexity theory [32] for CP problem (7), the number of iterations to find an $\epsilon$-solution i.e., an approximate solution $x \in X$ such that $f(x) - f^* \leq \epsilon$, does not exceed $O(1/\epsilon^2)$, if $f$ is a general non-smooth Lipschitz continuous convex function. For smooth convex optimization, the optimal iteration complexity bound is $O(1/\epsilon)$. Furthermore, in case that $f$ is weakly smooth and its gradient is Hölder continuous, the optimal iteration complexity is bounded by $O\left(1/\epsilon^{\frac{2}{\nu}}\right)$ for some $\rho \in (0, 1)$. It follows from Theorem 1 that the iteration complexity bound of the MAPL method is

$$O\left(\frac{MD_0^{\frac{1}{2}+\rho}}{\epsilon}\right)^{\frac{2}{\nu}}.$$ 

In other words, the MAPL method can achieve uniformly optimal iteration complexity bounds for solving non-smooth ($\rho = 0$), smooth ($\rho = 1$) and weakly smooth ($\rho \in (0, 1)$) CP problems.

4. The Modified Uniform Smoothing Level (MUSL) Method

In this section we consider the objective function $f$ in (7) with the form of

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

where $\hat{f} : X \to \mathbb{R}$ is Lipschitz continuous and simple. In addition, $F(x)$ has a special structure that

$$F(x) := \max_{y \in Y} \{ \langle Ax, y \rangle - \hat{g}(y) \}. \quad (33)$$

Here, the compact convex set $Y \subset \mathbb{R}^n$ is nonempty, $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator and $\hat{g} : Y \to \mathbb{R}$ is convex and continuous on $Y$.

Generally, the function $F$ is convex and non-smooth. In this case, $F$ can be approximated by constructing a series of smooth convex functions [16,17]. Let $v(y)$ denote the prox-function of compact convex set $Y$ with coefficient $c_v$. $c_v := \arg \min_{y \in Y} v(y)$ refers to the prox-center of $v(y)$. Let

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

The functions $F(x)$ and $f(x)$ are approximated by $F_\eta(x)$ and $f_\eta(x)$ (with some smoothing parameter $\eta > 0$), respectively, i.e.,

$$F_\eta(x) := \max_{y \in Y} \{ \langle Ax, y \rangle - \hat{g}(y) - \eta V(y) \},$$

$$f_\eta(x) := \hat{f} + F_\eta(x). \quad (34)$$

As described in [17], from the first-order optimality conditions, the convexity of $\hat{g}(y)$ and the strong convexity of $V(y)$, we know that the gradient of $F_\eta(x)$ is Lipschitz continuous with Lipschitz constant

$$L_\eta \equiv L(F_\eta) := \frac{||A||^2}{\eta c_v}. \quad (35)$$

Furthermore, we have

$$F_\eta \leq F(x) \leq F_\eta + \eta D_{c_v,Y}, \quad (36)$$

and

$$f_\eta \leq f(x) \leq f_\eta + \eta D_{c_v,Y}, \quad (37)$$

for any $x \in X$.

Inspired by the smoothing technique of [16,17], Lan [24] proposed the uniform smoothing level (USL) method for solving structure non-smooth CP problem (7) with $f$ being defined by (32).
The advantage of the USL method is that the parameter $\eta$ and estimation of $D_{v,g}$ can be automatically adjusted and obtained during the gap reduction procedure $G_{USL}$, which makes the USL method free of parameters. However, similar to the APL method, each iteration of the USL method also involves two subproblems. Based on the USL method, combining with the analysis of the MAPL method in the previous section, we propose the modified uniform smoothing level (MUSL) method. Same as the USL method, the MUSL method can achieve the optimal iteration complexity when solving problem (7) with (32), but only one subproblem is required to solve in each iteration.

We next describe the gap reduction procedure $G_{MUSL}$ of MUSL method. As the internal procedure of MUSL method, $G_{MUSL}$ is called to compress the gap between upper and lower bounds of $f^*$.

**The MUSL gap reduction procedure:** $(p^+, \text{lb}^+, \bar{D}^+) = G_{MUSL}(p, \text{lb}, \bar{D}, \lambda, \theta)$

**Step 0:** (Initialization) Set $x_0^0 := p$, $f_0 := f(x_0^0)$, $\text{lb}_0 := \text{lb}$ and $\ell := \lambda f_0 + (1-\lambda)\bar{f}_0$. Let $\eta := \theta(f_0 - \ell)/(2\bar{D})$. Choose $x_0 \in X$ and the initial localizer $X'_0 = X$. The prox-function $d_{v}(x)$ is defined in (15). Let $k = 1$.

**Step 1:** (Update level set) Let

\begin{align*}
x_k^1 &:= (1 - a_k) \hat{x}_{k-1}^1 + a_k x_{k-1}, \quad (39) \\
h_{\eta}(x^1_k, x) &:= f(x) + f_0(x_k^1) + \langle \nabla f_0(x_k^1), x - x_k^1 \rangle, \quad (40) \\
X_k &:= \{ x \in X'_{k-1} : h(x_k^1, x) \leq \ell \}. \quad (41)
\end{align*}

**Step 2:** (Update prox-center and lower bound) Let

\begin{equation}
 x_k = \text{argmin}_{x \in X_k} d_{v}(x). \quad (42)
\end{equation}

If $X_k = \emptyset$, then stop the procedure and output $p^+ = x_{k-1}^1$, $\text{lb}^+ = \ell$, $\bar{D}^+ = \bar{D}$.

**Step 3:** (Update upper bound) Let

\begin{align*}
\hat{x}_k^1 &= a_k x_k + (1-a_k) \hat{x}_{k-1}^1, \quad (43) \\
x_k^{\ell} &= \begin{cases} 
\hat{x}_k^1, & \text{if } f(\hat{x}_k^1) < f_{k-1}, \\
x_{k-1}^1, & \text{otherwise},
\end{cases}
\end{align*}

and $\bar{f}_k := f(x_k^{\ell})$. Check the following two possible stopping rules:

- **(3a)** if $\bar{f}_k \leq \ell + \theta(f_0 - \ell)$, stop the procedure with $p^+ = x_k^{\ell}$, $\text{lb}^+ = \text{lb}$, $\bar{D}^+ = \bar{D}$.
- **(3b)** Otherwise, if $f_0(\hat{x}_k^1) \leq \ell + \frac{\theta}{2}(f_0 - \ell)$, stop the procedure with $p^+ = x_k^{\ell}$, $\text{lb}^+ = \text{lb}$, $\bar{D}^+ = 2\bar{D}$.

**Step 4:** (Update localizer) Choose an arbitrary $X'_k$ such that $X_k \subseteq X'_k \subseteq X_k$, where

\begin{equation}
X_k = \{ x \in X : \langle \nabla d_{v}(x_k), x - x_k \rangle \geq 0 \}. \quad (44)
\end{equation}

**Step 5:** (Loop) Set $k = k + 1$ and return to Step 1.

The following are a few remarks to the procedure $G_{MUSL}$ including the differences from $G_{MAPL}$ and some properties. Firstly, compared with $G_{MAPL}$, the procedure $G_{MUSL}$ needs to input one more parameter $\bar{D}$ which is used to calculate the smoothing parameter $\eta$, so that the approximate function $f_{\eta}$ of $f$ can be defined. Secondly, these two procedures approximate the objective function in different ways. $G_{MAPL}$ approximates the objective function $f$ by the linearization of $f$ (18), while $G_{MUSL}$ only
linearizes the function $F_\eta$ and approximates the function $f$ with $h_\eta$ (40). According to (40), convexity of $F_\eta$, (36) and (32), we have

$$h_\eta(x_k^l, x) \leq \hat{f}(x) + F_\eta(x) \leq \hat{f}(x) + F(x) = f(x), \forall x \in X,$$

(45)

which means that the function $h_\eta(x_k^l, x)$ is the lower estimate of $f(x)$. Thirdly, the procedure $G_{\text{MUSL}}$ has one more exit than $G_{\text{MAPL}}$ with three possible exits: Step 2, Step 3a, and Step 3b. If the procedure stops at Step 2, the lower bound $lb$ is updated to $\ell$; if it stops at Step 3a, we say that a significant improvement has been made on the upper bound of $f^*$ and update the upper bound $\mathcal{U}_{lb}$. In addition, if the procedure stops at Step 3b, it is considered that there is no significant improvement on the upper bound of $f^*$, thus the parameter $D$ needs to be adjusted to estimate $D_{v,Y}$ and we update $D$ to $2D$.

Referring to the work of [24,31], the following lemma gives some simple observations of procedure $G_{\text{MUSL}}$. Its proof can be found in Appendix A.

**Lemma 2.** The following results hold for internal procedure $G_{\text{MUSL}}$:

a. If $G_{\text{MUSL}}$ terminates at Step 2 or Step 3a, we have $f(p^+) - lb^+ \leq q(f(p) - lb)$, where $q \equiv q(\lambda, \theta) = \max\{1, 1 - \lambda(1 - \theta)\}$;

b. If $G_{\text{MUSL}}$ terminates at Step 3b, we have $D < D_{v,Y}$ and $D^+ < 2D_{v,Y}$.

Similar to the previous section, it is time to build the convergence results for procedure $G_{\text{MUSL}}$. Referring to the theoretical analysis mode of [24], the detailed derivations of the iteration complexity are given as follows.

**Proposition 2.** Let $\{a_k\}$ satisfy (24) and take $\gamma = 1$. Let $(x_k^l, x_k, x_k^u) \in X \times X \times X, k \geq 1$ be the search points, $\ell$ be the level parameter, $d_{\omega}(\cdot)$ be the prox-function and $\eta$ be the smoothing parameter. Then the number of iterations of internal procedure $G_{\text{MUSL}}$ can be bounded by,

$$N(D_0, D) := \left[\frac{2\|A\|}{\lambda D_0} \sqrt{\frac{D_{\omega,X}D}{c_\omega c_v}}\right],$$

(46)

where $D_0 = ub - lb = \mathcal{U}_{lb} - f_{\text{lb}}$. $D_{\omega,X}$ is defined in (14).

**Proof.** Suppose the gap reduction procedure $G_{\text{MUSL}}$ does not stop at the $K$-th iteration. From (34) and inequality (45), we have $h_\eta(z, x) \leq f_\eta(x), \forall z, x \in X$. Because of (34), (40) and the Lipschitz continuity of the subgradient of $F_\eta$, we have

$$f_\eta(x) - h_\eta(z, x) = F_\eta(x) - [F_\eta(z) + (\nabla F_\eta(z), x - z)] \leq \frac{L_\eta}{2}\|x - z\|^2, \forall z, x \in X.$$

Hence,

$$f_\eta(x) \leq h_\eta + \frac{L_\eta}{2}\|x - z\|^2, \forall x, z \in X.$$

From (40), the definition of $x_k^u$ (43), the convexity of $\hat{f}$, (45) and (44), we have for all $k \geq 1$,

$$h_\eta(x_k^l, x_k^u) = \hat{f}(x_k^u) + F_\eta(x_k^l) + \langle F_\eta'(x_k^l), x_k^u - x_k^l \rangle$$

$$= \hat{f}(a_k x_k + (1 - a_k)x_{k-1}^u) + F_\eta(x_k^l) + \langle F_\eta'(x_k^l), a_k x_k + (1 - a_k)x_{k-1}^u - x_k^l \rangle$$

$$\leq (1 - a_k) h_\eta(x_k^l, x_{k-1}^u) + a_k h_\eta(x_k^l, x_k)$$

$$\leq (1 - a_k) f_\eta(x_{k-1}^u) + a_k h_\eta(x_k^l, x_k)$$

$$\leq (1 - a_k) f_\eta(x_{k-1}^u) + a_k \ell.$$. 
Moreover, by (39) and (43) we can easily obtain
\[ \tilde{x}_k^l - x_k^l = a_k x_k + (1 - a_k) \tilde{x}_{k-1}^l - a_k x_{k-1} - (1 - a_k) \tilde{x}_{k-1}^l = a_k (x_k - x_{k-1}). \]

It then follows that
\[ f_\eta(\tilde{x}_k^l) \leq (1 - a_k) f_\eta(\tilde{x}_{k-1}^l) + a_k \ell + \frac{L_\eta a_k^2}{2} \|x_k - x_{k-1}\|^2. \]

Subtracting both sides of the above inequality by \( \ell \), then dividing both sides by \( a_k^2 \), we have
\[ \frac{f_\eta(\tilde{x}_k^l) - \ell}{a_k^2} \leq \frac{1 - a_k}{a_k^2} [f_\eta(\tilde{x}_{k-1}^l) - \ell] + \frac{L_\eta}{2} \|x_k - x_{k-1}\|^2. \]

Summing up the above inequalities over \( k \), together with (24) and Hölder inequality, we obtain
\[ f_\eta(\tilde{x}_k^l) - \ell \leq \frac{2L_\eta d_\omega(x_k)}{\sigma_\omega K^2} \leq \frac{c^2 L_\eta d_\omega(x_k)}{\sigma_\omega K^2}. \]

Thus, from the relation \( d_\omega(x_k) \leq D_{\omega,X} \) (35) and (38), we have
\[ f_\eta(\tilde{x}_k^l) - \ell \leq \frac{L_\eta a_k^2}{2} \sum_{k=1}^K \|x_k - x_{k-1}\|^2 \leq \frac{c^2 L_\eta d_\omega(x_k)}{\sigma_\omega K^2}. \]

In conclusion, we obtain
\[ K < \frac{2c \|A\|}{\lambda \theta \Delta_0} \sqrt{\frac{D_{\omega,X} D}{\sigma_\omega \sigma_v}}. \]

Based on the results of the convergence of the procedure \( G_{MUSL} \) above, we next give the algorithm of the MUSL method. Similar to the MAPL method, the MUSL method is also implemented with an external algorithm framework and internal gap reduction procedure \( G_{MUSL} \). The external algorithm of the MUSL method is mainly to determine whether the gap between the upper and lower bounds on \( f^* \) has reached the given tolerance \( \epsilon \) in current iteration \( k \). If the given tolerance \( \epsilon \) is reached, the algorithm terminates and output an approximate optimal solution of \( f \), otherwise the external algorithm call the internal procedure \( G_{MUSL} \) continuously to compress the gap between the upper and lower bounds. The algorithmic framework of the MUSL method is as follows (Algorithm 3).
Then, we have the following relation

\[ F(x_0) - F(x_1) - \langle F'(x_1), x_0 - x_1 \rangle \leq 2 \left( \frac{2\|A\|^2D_{v,Y}}{\sigma_v} \right)^{\frac{1}{2}} \|x_0 - x_1\|, \forall x_0, x_1 \in \mathbb{R}^n. \]  

Here \( F'(x_1) \in \partial F(x_1) \) denotes the subgradient of \( F \) at \( x_1 \) and \( D_{v,Y} \) is defined in (13).

**Lemma 3.** Suppose that \( F \) is defined in (33) and \( v \) is a prox-function of \( Y \) with strong convex modulus \( \sigma_v \). Then, we have the following relation

\[ F(x_0) - F(x_1) - \langle F'(x_1), x_0 - x_1 \rangle \leq 2 \left( \frac{2\|A\|^2D_{v,Y}}{\sigma_v} \right)^{\frac{1}{2}} \|x_0 - x_1\|, \forall x_0, x_1 \in \mathbb{R}^n. \]  

We now turn to analyze the optimal complexity bound for the MUSL method. Please note that the following results are modifications of those in [24].

**Theorem 2.** Suppose that \( \{\alpha_k\} \) in procedure \( G_{\text{MUSL}} \) is chosen to satisfy condition (24), and that \( \gamma = 1 \). Then for given \( \epsilon > 0 \), the following statements hold for the MUSL method.

1. The number of non-significant phases can be bounded by

\[ S_1(\tilde{D}_1) := \left[ \log_2 \frac{D_{v,Y}}{\tilde{D}_1} \right], \]

and the number of significant phases can be bounded by

\[ S_2(\epsilon) := \max \left\{ 0, \log_\frac{1}{\epsilon} \left[ 2\frac{\|A\|\Omega_{\omega,X}^{1/2}\Omega_{v,Y}^{1/2}}{\epsilon} \right] \right\}. \]

2. The total number of iterations performed by the MUSL method does not exceed

\[ S_1(\tilde{D}_1) + S_2(\epsilon) + \frac{2\Omega_{\omega,X}^{1/2}\Omega_{v,Y}^{1/2}}{\epsilon}, \]

where \( \epsilon = \frac{\|A\|}{\|F\|} (\sqrt{2}/(1-q) + \sqrt{2} + 1) \), and \( \Omega_{\omega,X} \) and \( \Omega_{v,Y} \) are defined in (14).

**Proof.** (1) Let \( \Delta_s = ub_s - lb_s \), \( s \geq 1 \). Without loss of generality, we suppose that \( \Delta_s \geq \epsilon \). From conclusion (2) of Lemma 2, we know that \( D_{S_1(\tilde{D}_1)+1} = 2D_{S_1(\tilde{D}_1)} \leq D_{v,Y} \), if non-significant phase occurs. From \( D_{S_1+1} = 2^{S_1(\tilde{D}_1)}\tilde{D}_1 \) and \( 2^{S_1(\tilde{D}_1)}\tilde{D}_1 < D_{v,Y} \), we have

\[ S_1(\tilde{D}) < \log_2 \frac{D_{v,Y}}{\tilde{D}_1}. \]
Furthermore, by (47) and (48) in Lemma 3 we have
\[
\Delta_1 = f(p_1) - h_0(p_0,p_1) = F(p_1) - [F(p_0) + \langle F'(p_0), p_1 - p_0 \rangle]
\leq 2\sqrt{2\|A\|^2 D_2 X} \left\| p_1 - p_0 \right\|
\leq 2\|A\| \sqrt{\frac{2D_{\omega,Y}}{\sigma_{\omega}} \frac{2D_{\omega,X}}{\sigma_{\omega}}}
= 2\|A\| \Omega_{\omega,X}^{\frac{1}{2}} \Omega_{\omega,Y}^{\frac{1}{2}}.
\]

From (e) in Lemma 1 and the definition of \(u_b\) and \(b_s\), we know that \(\varepsilon < \Delta S_2(\varepsilon) + 1 \leq q \Delta S_2(\varepsilon) \leq q S_2(\varepsilon) \Delta_1\) and \((\frac{1}{q}) S_2(\varepsilon) \leq \frac{\Delta_1}{\varepsilon}\). In conclusion, we obtain that
\[
S_2(\varepsilon) \leq \log_2 \frac{\Delta_1}{\varepsilon} \leq \log_2 \frac{2\|A\| \Omega_{\omega,X}^{\frac{1}{2}} \Omega_{\omega,Y}^{\frac{1}{2}}}{\varepsilon}.
\]

(2) Let \(K_1 = \{a_1, a_2, ..., a_m\}\) and \(K_2 = \{b_1, b_2, ..., b_m\}\) denote the index sets of non-significant and significant phases respectively. For any \(1 \leq i \leq m\) in the non-significant phases, we have \(\tilde{D}_{a_{i+1}} = 2\tilde{D}_{a_i}\). Then we know that \(\tilde{D}_{b_i} = 2^{i-1}\tilde{D}_{b_i} = 2^{i-1}\tilde{D}_{1}\) and \(\Delta_{b_i} > \varepsilon\). Due to Proposition 2, we know that \(K\) decreases about the first variable and increases about the second variable monotonously. In addition, we conclude that the total number of iterations performed by the significant phases can be bounded by
\[
N_1 = \sum_{i=1}^{m} K_{\text{MUSL}}(\Delta_{b_i}, \tilde{D}_{b_i}) \leq \sum_{i=1}^{m} K_{\text{MUSL}}(\varepsilon, \tilde{D}_{b_i})
\leq m + \sum_{i=1}^{m} \frac{2\varepsilon\|A\|}{\lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} \tilde{D}_{b_i}}{\sigma_{\omega} \sigma_{\varepsilon}}}
\leq m + \sum_{i=1}^{m} \frac{2\varepsilon\|A\|}{\lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} 2^{i-1} \tilde{D}_{b_i}}{\sigma_{\omega} \sigma_{\varepsilon}}}
\leq S_1(\tilde{D}_1) + \frac{2\varepsilon\|A\|}{(\sqrt{2} - 1) \lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} D_{\omega,Y}}{\sigma_{\omega} \sigma_{\varepsilon}}}
\leq S_1(\tilde{D}_1) + \frac{2\varepsilon\|A\|}{(\sqrt{2} - 1) \lambda \theta \varepsilon} \sqrt{\Omega_{\omega,X}^{\frac{1}{2}} \Omega_{\omega,Y}^{\frac{1}{2}}}
\leq S_1(\tilde{D}_1) + \frac{\varepsilon\|A\| \Omega_{\omega,X}^{\frac{1}{2}} \Omega_{\omega,Y}^{\frac{1}{2}}}{(\sqrt{2} - 1) \lambda \theta \varepsilon}.
\]

From the fact that \(\Delta_{b_{j+1}} \leq q \Delta_{b_j}, \forall 1 \leq j \leq n\), we know that
\[
\Delta_{b_j} \geq q^{j-n} \varepsilon \Delta_{b_n} > q \varepsilon^{j-n}.
\]
It follows from the monotonic of $K$ about variables and Proposition 2 that $\tilde{D}_h \leq 2D_{v,Y}$. In addition, we obtain that the total number of iterations performed in the non-significant phases can be bounded by

$$N_2 = \sum_{j=1}^{n} K_{\text{MUSL}}(\Delta_{h_j}, \tilde{D}_h) \leq \sum_{j=1}^{n} K_{\text{MUSL}}(eq^{-n}, 2D_{v,Y})$$

$$\leq n + \sum_{j=1}^{n} \frac{2c\|A\|}{\lambda \theta e q^{-n}} \sqrt{\frac{D_{v,X}2D_{v,Y}}{\sigma_v \sigma_0}}$$

$$\leq n + \frac{2c\|A\|}{\lambda \theta e} \sqrt{\frac{2D_{v,X}D_{v,Y}}{\sigma_v \sigma_0} \sum_{j=1}^{n} q^{n-j}}$$

$$\leq S_2(e) + \frac{2c\|A\|}{(1-q)\lambda \theta e} \sqrt{\frac{2D_{v,X}D_{v,Y}}{\sigma_v \sigma_0} \sum_{j=1}^{n} q^{n-j}}$$

$$= S_2(e) + \frac{\sqrt{2}c\|A\|\Omega_{v,X}^{1/2} \Omega_{v,Y}^{1/2}}{(1-q)\lambda \theta e}.$$

Combining (49) and (50), we conclude that the total number of iterations of the MUSL method does not exceed

$$N_1 + N_2 = S_1(\tilde{D}_1) + \frac{c\|A\|\Omega_{v,X}^{1/2} \Omega_{v,Y}^{1/2}}{(\sqrt{2} - 1)\lambda \theta e} + S_2(e) + \frac{\sqrt{2}c\|A\|\Omega_{v,X}^{1/2} \Omega_{v,Y}^{1/2}}{(1-q)\lambda \theta e}$$

$$= S_1(\tilde{D}_1) + S_2(e) + \left(\frac{1}{1-q} + \frac{\sqrt{2}}{1-q}\right) \frac{c\|A\|\Omega_{v,X}^{1/2} \Omega_{v,Y}^{1/2}}{\lambda \theta e}.$$

□

In what follows, we apply the MUSL method to solve a special problem where $\hat{f}$ in (32) is a smooth convex function. In addition, the complexity results of procedure $G_{\text{MUSL}}$ and MUSL method are established in this special case. $\hat{f}$ is a smooth convex function means that there exists a constant $L_1 > 0$ such that $\hat{f}$ satisfies

$$\hat{f}(z) - \hat{f}(x) - \langle \nabla \hat{f}(x), z - x \rangle \leq \frac{L_1}{2} \| z - x \|^2, \forall z, x \in X. \quad (51)$$

From the fact that $F_{\eta}$ has a Lipschitz continuous gradient with Lipschitz-constant $L_{\eta} > 0$, we have

$$F_{\eta}(z) - F_{\eta}(x) - \langle \nabla F_{\eta}(x), z - x \rangle \leq \frac{L_{\eta}}{2} \| z - x \|^2, \forall z, x \in X.$$

Thus, we know that $f_{\eta} := \hat{f}(x) + F_{\eta}(x)$ is a smooth function on $X$ and satisfies

$$f_{\eta} - f_{\eta} - \langle \nabla f_{\eta}, z - x \rangle \leq \frac{L}{2} \| z - x \|^2, \forall z, x \in X,$$

where $L = L_1 + L_{\eta}$. From the fact that $F$ is non-smooth, we know that $f$ is non-smooth as well.

We make the following changes to the outer algorithm of the MUSL method and internal procedure $G_{\text{MUSL}}$. Replace (47) in outer algorithm with

$$p_1 \in \text{argmin}_{x \in X} \{ h_0(p_0, x) := f_{\eta}(p_0) + \langle \nabla f_{\eta}(p_0), x - p_0 \rangle \},$$

(53)
and (40) in procedure $G_{\text{MUSL}}$ with
\[ h_\eta(x_k^l, x) := f_\eta(x) + f_\eta(x_k^l) + \langle \nabla f_\eta(x_k^l), x - x_k^l \rangle. \]

The function $f_\eta$ is fixed as a result of the fact that $\eta$ is fixed in the internal procedure $G_{\text{MUSL}}$. Similar to Proposition 2, we have the following results.

**Theorem 3.** We assume that \( \{\alpha_k\} \) satisfies condition (24) and take $\gamma = 1$. Let $\ell$ be the level parameter, $d_\omega(\cdot)$ denotes the prox-function and $\eta$ is the smoothing parameter in procedure $G_{\text{MUSL}}$. If $\hat{f}$ is a smooth function, then the number of iterations of procedure $G_{\text{MUSL}}$ can be bounded by,

\[
\mathcal{N} := c \Omega^{\frac{1}{\omega}, X} \left( \sqrt{\frac{L_1}{\theta \Delta_0}} + \sqrt{2} \| A \| \sqrt{\frac{\tilde{D}}{\sigma_{v,Y}}} \right) + 1,
\]

where $\Delta_0 = \text{ub} - \text{lb} = f_0 - f_{\ell_0}$, and $\Omega^{\omega, X}$ is defined in (14).

**Proof.** Suppose that the procedure $G_{\text{MUSL}}$ does not stop in iteration $K$. Because of the features of $\hat{f}$ and $F_\eta$, we know that $f_\eta$ satisfies (52). Let $L_\eta = L$. From (14), we have

\[
f_\eta(s_K^\mu) - \ell \leq c 2 \Omega^{\omega, X} \frac{\lambda \theta \Delta_0}{2K^2}, \forall K \geq 1.
\]

Then according to the stopping test in Step 3b in procedure $G_{\text{MUSL}}$, we have

\[
f_\eta(s_K^\mu) - \ell > \frac{2 \lambda \theta \Delta_0}{2}.
\]

Due to $L = L_1 + L_\eta$, (35) and (38) we obtain that

\[
K < c \Omega^{\omega, X} \left( \sqrt{\frac{L_1}{\theta \Delta_0}} + \sqrt{2} \| A \| \sqrt{\frac{\tilde{D}}{\sigma_{v,Y}}} \right) = c \Omega^{\omega, X} \left( \sqrt{\frac{L_1}{\theta \Delta_0}} + \sqrt{2} \| A \| \sqrt{\frac{\tilde{D}}{\sigma_{v,Y}}} \right).
\]

Thus, we have

\[
\mathcal{N} = c \Omega^{\omega, X} \left( \sqrt{\frac{L_1}{\theta \Delta_0}} + \sqrt{2} \| A \| \sqrt{\frac{\tilde{D}}{\sigma_{v,Y}}} \right) + 1.
\]

\(\square\)

Here we will also establish the complexity of the MUSL method under this special case where $\hat{f}$ is smooth and convex.

**Theorem 4.** Assume that $\{\alpha_k\}$ satisfy condition (24) and take $\gamma = 1$. If $\hat{f}$ is a smooth convex function, then the following statements hold for the MUSL method.

1. The number of non-significant phases can be bounded by

   \[
   \mathbb{S}_1 := \left\lceil \log_2 \frac{D_{v,Y}}{D_1} \right\rceil,
   \]
and the number of significant phases can be bounded by

\[ \mathbb{S}_2 := \max \left\{ 0, \log_{\frac{1}{\sqrt{2}}} \frac{2\|A\| \Omega_{\omega,X}^\frac{1}{2} \Omega_{\omega,Y}^\frac{1}{2}}{\varepsilon} \right\}. \]

(2) the total number of iterations performed by the MUSL method does not exceed

\[ \mathbb{S}_1 + \mathbb{S}_2 + \left( \mathbb{S}_1 + \frac{1}{1 - \sqrt{q}} \right) c\Omega_{\omega,X}^\frac{1}{2} \sqrt{\frac{L_1}{\lambda \theta \varepsilon}} + \left( \frac{1}{\sqrt{2} - 1} + \frac{\sqrt{q}}{1 - q} \right) c\|A\| \Omega_{\omega,X}^\frac{1}{2} \Omega_{\omega,Y}^\frac{1}{2}. \]

**Proof.** Let \( \Delta_s = \log_{\frac{1}{\sqrt{2}}} \frac{D_{\omega,Y}}{D_{\hat{s}}}. \) By (53), (48) and (51) we have

\[ \mathbb{S}_1 \leq \|A\| \sqrt{\frac{D_{\omega,Y} L_1}{\sigma_{\omega} \sigma_{\omega}}} + \frac{L_1}{2} D_{\omega,X}. \]

From (e) in Lemma 1 and the definition of \( \text{ub}_s \) and \( \text{lb}_s \), we know that \( \varepsilon < \Delta_{s+1} \leq q \Delta_s \leq q^s \Delta_1 \) and \( \left( \frac{1}{q} \right)^s \leq \Delta_1 \). In conclusion, we obtain that

\[ \mathbb{S}_2 \leq \log_{\frac{1}{\sqrt{2}}} \frac{\Delta_1}{\varepsilon} \leq \log_{\frac{1}{\sqrt{2}}} \frac{2\|A\| \Omega_{\omega,X}^\frac{1}{2} \Omega_{\omega,Y}^\frac{1}{2} f_{\hat{s}}}{\lambda \theta \varepsilon} + \frac{L_1}{2} D_{\omega,X}. \]

(2) Let \( \mathcal{K}_1 = \{ c_1, c_2, ..., c_m \} \) and \( \mathcal{K}_2 = \{ d_1, d_2, ..., d_m \} \) denote the index sets of non-significant and significant phases respectively. For any \( 1 \leq i \leq m \) in the non-significant phases, we have \( Q_{c_i+1} = 2Q_{c_i} \). Then we know that \( Q_{c_i} = 2^{i-1} Q_{c_1} = 2^{i-1} Q_1 \) and \( \Delta_{c_i} > \varepsilon \). By Proposition 2, we know that \( \mathcal{K} \) decreases about the first variable and increases about the second variable monotonously. It follows that the total number of iterations performed by the significant phases can be bounded by

\[ \mathcal{N}_1 = \sum_{i=1}^{m} K_{\text{MUSL}} (\Delta_{c_i}, D_{c_i}) \leq \sum_{i=1}^{m} K_{\text{MUSL}} (\varepsilon, D_{c_i}) \]

\[ \leq m + \sum_{i=1}^{m} \frac{2c\|A\|}{\lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} D_{c_i}}{\sigma_{\omega} \sigma_{\omega}}} \sqrt{\frac{D_{\omega,Y} L_1}{\sigma_{\omega} \lambda \theta \varepsilon}} + 2c\|A\| \sqrt{\frac{D_{\omega,X} D_{c_i}}{\sigma_{\omega} \sigma_{\omega}}} \sum_{i=1}^{m} 2^{i-1} \]

\[ \leq \mathbb{S}_1 \left( 1 + c \sqrt{\frac{2D_{\omega,X} L_1}{\sigma_{\omega} \lambda \theta \varepsilon}} \right) + \frac{2c\|A\|}{\lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} D_{\omega,Y}}{\sigma_{\omega} \sigma_{\omega}}} \sum_{i=1}^{m} 2^{i-1} \]

\[ \leq \mathbb{S}_1 \left( 1 + c \sqrt{\frac{2D_{\omega,X} L_1}{\sigma_{\omega} \lambda \theta \varepsilon}} \right) + \frac{2c\|A\|}{\lambda \theta \varepsilon} \sqrt{\frac{D_{\omega,X} D_{\omega,Y}}{\sigma_{\omega} \sigma_{\omega}}} \]

\[ = \mathbb{S}_1 \left( 1 + c \Omega_{\omega,X}^\frac{1}{2} \sqrt{\frac{L_1}{\lambda \theta \varepsilon}} \right) + \frac{c\|A\|}{\lambda \theta \varepsilon} \Omega_{\omega,X}^\frac{1}{2} \Omega_{\omega,Y}^\frac{1}{2}. \]
From the fact that \( \Delta_{d_{i1}} \leq q \Delta_{d_j} \forall 1 \leq j \leq n \), we know that
\[
\Delta_{d_j} \geq q^{j-n} \Delta_{d_n} > \epsilon q^{-n}.
\]

By the monotonicity of \( K \) about variables and the fact that \( \tilde{D}_{b_j} \leq 2 D_{b,Y} \), we know that the total number of iterations performed in the non-significant phases can be bounded by
\[
\overline{N}_2 = \sum_{j=1}^{n} K_{\text{MUSL}} \left( \Delta_{d_j}, \tilde{D}_{b_j} \right) \leq \sum_{j=1}^{n} K_{\text{MUSL}} \left( \epsilon q^{-n}, 2 D_{b,Y} \right)
\]
\[
\leq n + \sum_{j=1}^{n} \frac{2c \|A\|}{\lambda \theta \epsilon} \sqrt{\frac{D_{b,Y} 2 D_{b,Y}}{\sigma_\omega \sigma_\nu}}
\]
\[
\leq n + \frac{2c \|A\|}{\lambda \theta \epsilon} \sum_{j=1}^{n} q^{n-j} \sqrt{\frac{2D_{b,Y} 2D_{b,Y}}{\sigma_\omega \sigma_\nu}}
\]
\[
\leq 3 + c \frac{1}{1 - \sqrt{q}} \sqrt{\frac{2D_{b,Y} L_1}{\sigma_\omega \lambda \theta \epsilon}} + \frac{2c \|A\|}{(1-q) \lambda \theta \epsilon} \sqrt{\frac{2D_{b,Y} 2D_{b,Y}}{\sigma_\omega \sigma_\nu}}
\]
\[
= 3 + c \frac{1}{1 - \sqrt{q}} \sqrt{\frac{L_1}{\lambda \theta \epsilon}} + \frac{\sqrt{2c \|A\| \Omega_{b,Y}^2 \Omega_{b,Y}^2}}{(1-q) \lambda \theta \epsilon}.
\] (55)

This together with (54) and (55) shows that the total number of MUSL method does not exceed \( \overline{N}_1 + \overline{N}_2 \). \( \Box \)

5. Two-Stage Stochastic Programming and Numerical Experiments

In this section, the MAPL and MUSL methods are applied to solve the two-stage stochastic programming problems. Furthermore, we compare the two modified methods with the APL and USL methods. All algorithms are implemented in MATLAB (R2014a), and Mosek (8.0) is called for solving subproblems. The environment in which the program runs is Windows 7 (64 bite), Intel(R) Core(TM) i7-6700 CPU 3.40GHz and 16G memory.

Consider the following two-stage stochastic programming with recourse
\[
\min \{ f(x) = c^T x + \mathbb{E}[Q(x, \xi)] : x \in X \},
\] (56)
where
\[
Q(x, \xi) = \min \{ q^T y : Wy = h + Tx, y \geq 0 \}.
\] (57)

Here, \( x \in \mathbb{R}^{n_1} \) and \( y \in \mathbb{R}^{n_2} \) are the decision variables of the first-stage problem and the second-stage problem respectively. \( X \neq \emptyset \) is a compact convex set, and \( \xi = (q, h, T) \in \mathbb{R}^{n_1 + n_2 + n_2 + n_1} \) is a random vector with a known probability distribution and support \( \Xi \). In [33], it was pointed out that by strong convexity one has
\[
Q(x, \xi) = \max \{ \pi^T (h + Tx) : W^T \pi \leq q \}.
\] (58)

Let \( \Pi(q) = \{ \pi \in \mathbb{R}^{n_2} : W^T \leq q \} \) be the feasible set of (58) and we assume that \( \Pi(q) \neq \emptyset, \forall q \in \Xi \).

It is easy to know that \( Q(x, \xi) \) is generally non-smooth. Therefore, for general distributions of \( \xi \), (56) is a non-smooth convex programming problem. In [33] Ahmed applied Nesterov’s smoothing technique to the two-stage stochastic programming problem (56), and established a proper smooth
approximation to $Q(x, \xi)$ as follows. For a given smoothing parameter $\mu$, consider the function $f_\mu(x)$ with the form

$$f_\mu(x) = c^T x + E(Q_\mu(x, \xi)),$$

(59)

where

$$Q_\mu(x, \xi) = \max\{ \pi^T (h + Tx) - \frac{\mu}{2} \pi^T \pi : \pi \in \Pi(q) \}.$$

(60)

For $\xi$ with discrete distribution, it is shown in \cite{34,35} that $f_\mu$ is differentiable and its gradient is Lipschitz continuous. Furthermore, when $\mu$ is sufficiently small, $f_\mu$ can approximate $f$ uniformly.

Based on the above analysis, we next carry out the MAPL and MUSL methods to solve the problem (59) and (60) to illustrate the effectiveness of the methods and compare them with the APL and USL methods.

We perform numerical experiments on some existing SP instances in \cite{1,36,37} including a telecommunication design (SSN) problem and a motor freight carrier routing problem (20Term). The SSN problem studied by Sen, Doverspike, and Cosares \cite{1} comes from the telecommunications industry: the first-stage problem allocates capacity between network links, and the second-stage problem generates demands to connections requested between point-to-point pairs. The 20Term problem studied by Mak, Morton, and Wood \cite{37} comes from a motor freight carrier’s model: the first-stage problem determines a program of carriers, and the second-stage problem adjusts the program according a multi-commodity network. The instances of SSN, 20Term and Storm are downloaded from the link: http://pwp.gatech.edu/guanghui-lan/computer-codes/. The dimensions related to these instances are shown in Table 1. $n_i$ denotes the number of constraints in the $i$-stage problem, and $m_i$ denotes the number of variable in the $i$-stage problem. In particular, we assume that the number of possible realizations of $\xi$ is fixed, i.e. $N = 50$ or 100. In this case, a total of five instances are tested. The integers in the brackets are the number of possible realizations. For given parameters $\lambda \in (0, 1), \theta \in (0, 1)$, tolerance $\varepsilon = 1.0e^{-6}$ and stepsize $a_k = 2/(k + 1)$ such that (24), we compare the number of iterations and CPU time of different methods. The results are shown in Tables 2–4.

| Table 1. Data about instances. |
|-----------------------------|
| $n_1$ | $m_1$ | $n_2$ | $m_2$ | $\dim(\xi)$ | $\text{ub}_1$ | $\Delta_1$ |
| SSN(50) | 89 | 1 | 706 | 175 | 86 | $2.352586 \times 10^2$ | $1.008000 \times 10^3$ |
| SSN(100) | 89 | 1 | 706 | 175 | 86 | $2.407279 \times 10^2$ | $9.878400 \times 10^2$ |
| 20Term(50) | 63 | 3 | 764 | 124 | 40 | $7.718617 \times 10^5$ | $1.804693 \times 10^7$ |
| 20Term(100) | 63 | 3 | 764 | 124 | 40 | $7.664142 \times 10^5$ | $1.801832 \times 10^7$ |
| Storm(50) | 121 | 185 | 1259 | 528 | 117 | $2.138447 \times 10^7$ | $6.370092 \times 10^6$ |
and modified uniform smoothing level (MUSL) methods, for uniformly solving the black-box problems and a class of structure non-smooth problems. In addition, both MAPL and MUSL methods have certain advantages in algorithm efficiency and solution time.

Methods can achieve optimal complexities respectively. To illustrate the effectiveness of the modified methods, they were then applied to solve the two-stage stochastic programming with recourse and CP problems and a class of structure non-smooth problems. In addition, both MAPL and MUSL methods have certain advantages in algorithm efficiency and solution time.

Table 2. MAPL and APL methods on SP instances.

| Prob  | Alg  | Iter | Ub              | δ      | Time  |
|-------|------|------|-----------------|--------|-------|
| SSN(50) | MAPL | 135  | 4.838073 x 10^6 | 9.042063 x 10^-7 | 63.71 |
|       | APL  | 140  | 4.838181 x 10^6 | 7.252755 x 10^-7 | 66.39 |
| SSN(100) | MAPL | 215  | 7.352607 x 10^6 | 9.748218 x 10^-6 | 233.95 |
|       | APL  | 263  | 7.352607 x 10^6 | 9.835275 x 10^-7 | 292.95 |
| 20Term(50) | MAPL | 281  | 2.549453 x 10^5 | 1.206150 x 10^-6 | 193.9 |
|       | APL  | 401  | 2.549453 x 10^5 | 1.168752 x 10^-6 | 282.83 |
| 20Term(100) | MAPL | 386  | 2.549453 x 10^5 | 8.772477 x 10^-7 | 519.40 |
|       | APL  | 401  | 2.532875 x 10^5 | 1.059493 x 10^-4 | 536.20 |
| Storm(50) | MAPL | 119  | 1.562059 x 10^7 | 1.093373 x 10^-6 | 87.59 |
|       | APL  | 401  | 1.547024 x 10^7 | 2.211075 x 10^-3 | 518.62 |

Table 3. MUSL and USL methods on SP instances.

| Prob  | Alg  | Iter | Ub              | δ      | Time  |
|-------|------|------|-----------------|--------|-------|
| SSN(50) | MUSL | 208  | 4.838072 x 10^6 | 3.003666 x 10^-6 | 206.81 |
|       | USL  | 322  | 4.899140 x 10^6 | 9.39798 x 10^-7 | 309.43 |
| SSN(100) | MUSL | 181  | 7.352608 x 10^6 | 1.686893 x 10^-6 | 351.11 |
|       | USL  | 190  | 7.352699 x 10^6 | 8.517813 x 10^-7 | 367.12 |
| 20Term(50) | MUSL | 401  | 2.549453 x 10^5 | 6.026059 x 10^-5 | 547.93 |
|       | USL  | 401  | 2.549453 x 10^5 | 8.844555 x 10^-4 | 522.41 |
| 20Term(100) | MUSL | 380  | 2.532875 x 10^5 | 2.171117 x 10^-6 | 1148.29 |
|       | USL  | 401  | 2.532875 x 10^5 | 4.172471 x 10^-6 | 1308.03 |
| Storm(50) | MUSL | 299  | 2.138447 x 10^7 | 1.087785 x 10^-6 | 437.11 |
|       | USL  | 401  | 1.546949 x 10^7 | 2.866387 x 10^-4 | 1084.46 |

Table 4. MAPL and MUSL methods on SP instances.

| Prob  | Alg  | Iter | Ub              | δ      | Time  |
|-------|------|------|-----------------|--------|-------|
| SSN(50) | MAPL | 159  | 4.838072 x 10^6 | 1.016148 x 10^-6 | 90.68 |
|       | MUSL | 175  | 4.838073 x 10^6 | 9.421333 x 10^-6 | 201.82 |
| SSN(100) | MAPL | 190  | 7.352608 x 10^6 | 6.397336 x 10^-6 | 217.63 |
|       | MUSL | 275  | 7.352607 x 10^6 | 2.900934 x 10^-6 | 638.96 |
| 20Term(50) | MAPL | 347  | 2.549453 x 10^5 | 4.465430 x 10^-6 | 271.94 |
|       | MUSL | 401  | 2.549453 x 10^5 | 3.039380 x 10^-5 | 663.75 |
| 20Term(100) | MAPL | 401  | 2.532875 x 10^5 | 3.203178 x 10^-2 | 607.93 |
|       | MUSL | 401  | 2.532875 x 10^5 | 1.512163 x 10^-4 | 1174.29 |
| Storm(50) | MAPL | 19   | 1.866006 x 10^7 | 1.028180 x 10^-6 | 18.08 |
|       | MUSL | 29   | 2.138447 x 10^7 | 1.799315 x 10^-6 | 61.25 |

6. Conclusions

In this paper, we presented two modified BL-type methods, the modified accelerated prox-level (MAPL) and modified uniform smoothing level (MUSL) methods, for uniformly solving the black-box CP problems and a class of structure non-smooth problems. In addition, both MAPL and MUSL methods can achieve optimal complexities respectively. To illustrate the effectiveness of the modified methods, they were then applied to solve the two-stage stochastic programming with recourse and numerical experiments were carried out. Finally, the numerical results shown that the MAPL and MUSL methods have certain advantages in algorithm efficiency and solution time.
Author Contributions: C.T. mainly contributed to the algorithm design and convergence analysis; B.H. mainly contributed to the convergence analysis and numerical results; and Z.W. mainly contributed to the algorithm design. All authors have read and agree to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation (11761013, 71861002) and Guangxi Natural Science Foundation (2018GXNSFFA281007) of China.

Acknowledgments: The authors would like to thank for the support funds.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Lemma 1. (a) By using mathematical induction, we first show part a). It is obvious that $X_f(\ell) \subseteq X_0$. Assume that $X_{f,k-1}$ is a localizer of $X_f(\ell)$ i.e., $X_f(\ell) \subseteq X_{f,k-1}$, for all $k \geq 1$. Then we have that $y \in X_{f,k-1}$, for all $y \in X_f(\ell)$. By the definition of $h$ and convexity of $f$ we know that $h(x_k^l y) \leq f(y) \leq \ell$, for any $y \in X_f(\ell)$. From the fact that $X_f(\ell) \subseteq X_k$ and $X_k \subseteq X_k'$ we have that $X_f(\ell) \subseteq X_k'$.

(b) It follows from $\bar{f}_k := \min \left\{ \bar{f}_{k-1}, f(\bar{x}_k^l) \right\} \leq \bar{f}_{k-1}$ in Step 3 of procedure $G_{MAPL}$ that $f^* \leq \bar{f}_k \leq \bar{f}_{k-1} \leq \cdots \leq \bar{f}_1 \leq \bar{f}_0$, for all $k \geq 1$.

(c) Form a) we have that $X_f(\ell) \subseteq X_k$. By the optimal condition of (20) we have that $-f(x_k) \in N(x_k)$. Then we obtain that $\langle \Delta d_w(x_k), y - x_k \rangle \geq 0$, for any $y \in X_k$. By the definition of $X_k$ we have $y \in X_k$ and further $X_k \subseteq X_k$, i.e., there exists a set $X_k'$ such that $X_k \subseteq X_k' \subseteq X_k$. In conclusion, we have $X_f(\ell) \subseteq X_k \subseteq X_k' \subseteq X_k$, for all $k \geq 1$.

(d) It follows from the fact that $X$ is a compact convex set and the definition of $X_k$ that $X_k$ is a compact convex set. Together with the strong convexity of $d_w$, we know that there is a unique optimal solution for (20) if $X_k \neq \emptyset$. Then we have $\ell < f(x)$, for any $x \in X$. Furthermore, we know that $\ell \leq f^*$.

(e) From b) we know that $f(p^+) \leq f(x_k^l) = f(p)$. We have lb$^+ = \ell$ if the procedure $G_{MAPL}$ stops at Step 2. From the fact that $\ell = \lambda b + (1 - \lambda)f(p)$ we have

$$f(p^+) - lb^+ \leq f(p) - [\lambda b + (1 - \lambda)f(p)] = \lambda (f(p) - lb).$$

We have lb$^+ = lb$ and $f(P^+) = f(x_k^l) \leq \ell + \theta(f(p) - \ell)$ if the procedure $G_{MAPL}$ stops at Step 3. From the fact that $\ell = \lambda b + (1 - \lambda)f(p)$, we have

$$f(p^+) = (1 - \theta)\ell + \theta f(p) - \ell \leq (1 - \theta)\ell + \theta f(p) - \ell = f(p) - lb.$$

\[\square\]

Proof of Lemma 2. (a) The proof of a) is the same as the proof of e) of Lemma 1.

(b) If the internal procedure $G_{MUASL}$ stops at Step 3, we have that $\bar{f}_k \geq \ell + \theta(\bar{f}_0 - \ell)$ and $f_\eta(\bar{x}_k^l) \leq \ell + \frac{\theta}{2}(\bar{f}_0 - \ell)$. Then we have that

$$f(\bar{x}_k^l) - f_\eta(\bar{x}_k^l) \geq \bar{f}_k - f_\eta(\bar{x}_k^l) > \frac{\theta}{2}(\bar{f}_0 - \ell).$$

Furthermore we know that

$$\eta \bar{D} = \frac{\theta}{2}(\bar{f}_0 - \ell) < f(\bar{x}_k^l) - f_\eta(\bar{x}_k^l) < \eta D_{v,Y}.$$

Then we have $D < D_{v,Y}$. Together with the definition $D^+$ in Step 3b, we know that $D^+ = 2D < 2D_{v,Y}$. \[\square\]
References

1. Sen, S.; Doverspike, R.D.; Cosares, S. Network planning with random demand. *Telecommun. Syst.* **1994**, *3*, 11–30. [CrossRef]
2. Yang, L.F.; Jian, J.B.; Wang, Y.Y.; Dong, Z.Y. Projected mixed integer programming formulations for unit commitment problem. *Int. J. Electr. Power Energy Syst.* **2015**, *68*, 195–202. [CrossRef]
3. Yang, L.F.; Jian, J.B.; Zhu, Y.N.; Dong, Z.Y. Tight Relaxation Method for Unit Commitment Problem Using Reformulation and Lift-and-Project. *IEEE Trans. Power Syst.* **2015**, *30*, 13–23. [CrossRef]
4. Yang, L.F.; Zhang, C.; Jian, J.B.; Meng, K.; Xu, Y.; Dong, Z.Y. A novel projected two-binary-variable formulation for unit commitment in power systems. *Appl. Energy* **2017**, *187*, 732–745. [CrossRef]
5. Wallace, S.W.; Ziemba, W.T. *Applications of Stochastic Programming*; Society for Industrial and Applied Mathematics and the Mathematical Programming Society: Philadelphia, PA, USA, 2005.
6. Kelley, J.E., Jr. The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 703–712. [CrossRef]
7. Veinott, A.F. The Supporting Hyperplane Method for Unimodal Programming. *Oper. Res.* **1967**, *15*, 147–152. [CrossRef]
8. Ruszczyński, A. A regularized decomposition method for minimizing a sum of polyhedral functions. *Math. Program.* **1986**, *35*, 309–333. [CrossRef]
9. Linderoth, J.; Wright, S. Decomposition Algorithms for Stochastic Programming on a Computational Grid. *Comput. Optim. Appl.* **2003**, *24*, 207–250. [CrossRef]
10. Lemaréchal, C. Nonsmooth optimization and descent methods. In *Research Report 78-4*; IIASA: Laxenburg, Austria, 1978.
11. Mifflin, R. A modification and an extension of Lemaréchal’s algorithm for nonsmooth minimization. In *Nondifferential and Variational Techniques in Optimization*; Springer: Berlin, Germany, 1982; pp. 77–90.
12. Kiwiel, K.C. An aggregate subgradient method for nonsmooth convex minimization. *Math. Program.* **1983**, *27*, 320–341. [CrossRef]
13. Kiwiel, K.C. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Program.* **1990**, *46*, 105–122. [CrossRef]
14. Ruszczyński, A.; Świątniewski, A. Accelerating the regularized decomposition method for two stage stochastic linear problems. *Eur. J. Oper. Res.* **1997**, *101*, 328–342. [CrossRef]
15. Lemaréchal, C.; Nesterov, Y.; Nemirovskii, A. New variants of bundle methods. *Math. Program.* **1995**, *69*, 111–147. [CrossRef]
16. Ben-Tal, A.; Nemirovski, A. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*; SIAM: Philadelphia, PA, USA, 2001.
17. Ben-Tal, A.; Nemirovski, A. Non-euclidean restricted memory level method for large-scale convex optimization. *Math. Program.* **2005**, *102*, 407–456. [CrossRef]
18. Richtárik, P. Approximate Level Method for Nonsmooth Convex Minimization. *J. Optim. Theory Appl.* **2012**, *152*, 334–350. [CrossRef]
19. Fischer, F.; Helmberg, C. A parallel bundle framework for asynchronous subspace optimization of nonsmooth convex functions. *SIAM J. Optim.* **2014**, *24*, 795–822. [CrossRef]
20. Kim, K.; Petra, C.G.; Zavala, V.M. An asynchronous bundle-trust-region method for dual decomposition of stochastic mixed-integer programming. *SIAM J. Optim.* **2019**, *29*, 318–342. [CrossRef]
21. Van Ackooij, W.; Frangioni, A. Incremental bundle methods using upper models. *SIAM J. Optim.* **2018**, *28*, 379–410. [CrossRef]
22. Iutzeler, F.; Malick, J.; de Oliveira, W. Asynchronous level bundle methods. *Math. Program.* **2019**, *149*, 1–45. [CrossRef]
23. Tang, C.; Jian, J.; Li, G. A proximal-projection partial bundle method for convex constrained minimax problems. *J. Ind. Manag. Optim.* **2019**, *15*, 757–774. [CrossRef]
24. Lan, G. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. *Math. Program.* **2015**, *149*, 1–45. [CrossRef]
25. Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN USSR* **1983**, *269*, 543–547.
26. Auslender, A.; Teboulle, M. Interior gradient and proximal methods for convex and conic optimization. *SIAM J. Optim.* **2006**, *16*, 697–725. [CrossRef]
27. Lan, G.; Lu, Z.; Monteiro, R.D. Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming. *Math. Program.* **2011**, *126*, 1–29. [CrossRef]
28. Lan, G. An optimal method for stochastic composite optimization. *Math. Program.* **2012**, *133*, 365–397. [CrossRef]
29. Nesterov, Y. *Introductory Lectures on Convex Optimization a Basic Course*; Springer Science & Business Media: New York, NY, USA, 2004; Volume 87.
30. Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.* **2005**, *103*, 127–152. [CrossRef]
31. Chen, Y.; Lan, G.; Ouyang, Y.; Zhang, W. Fast Bundle-Level Type Methods for Unconstrained and Ball-Constrained Convex Optimization. *Comput. Optim. Appl.* **2019**, *73*, 159–199. [CrossRef]
32. Nemirovsky, A.S.; Yudin, D. Problem Complexity and Method Efficiency in Optimization. In *Wiley-Interscience Series in Discrete Mathematics*; Wiley-Interscience: New York, NY, USA, 1983.
33. Ahmed, S. *Smooth Minimization of Two-Stage Stochastic Linear Programs*; Georgia Institute of Technology: Atlanta, GA, USA, 2006.
34. Chen, X.; Qi, L.; Womersley, R.S. Newton’s method for quadratic stochastic programs with recourse. *J. Comput. Appl. Math.* **1995**, *60*, 29–46. [CrossRef]
35. Chen, X. A parallel BFGS-SQP method for stochastic linear programs. In *Computational Techniques and Applications*; World Scientific: Princeton, NJ, USA, 1995; pp. 67–74.
36. Linderoth, J.; Shapiro, A.; Wright, S. The empirical behavior of sampling methods for stochastic programming. *Ann. Oper. Res.* **2006**, *142*, 215–241. [CrossRef]
37. Mak, W.; Morton, D.P.; Wood, R.K. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Oper. Res. Lett.* **1999**, *24*, 47–56. [CrossRef]