Stochastic Frank-Wolfe for Composite Convex Minimization

Francesco Locatello[*]   **Alp Yurtsever**[†]   **Olivier Fercoq**[‡]   **Volkan Cevher**[†]

```
francesco.locatello@inf.ethz.ch
{alp.yurtsever,volkan.cevher}@epfl.ch
olivier.fercoq@telecom-paristech.fr
```

[*]Department of Computer Science, ETH Zurich, Switzerland
[†]LIONS, Ecole Polytechnique Fédérale de Lausanne, Switzerland
[‡]LTCI, Télécom Paris, Université Paris-Saclay, France

Abstract

A broad class of convex optimization problems can be formulated as a semidefinite program (SDP), minimization of a convex function over the positive-semidefinite cone subject to some affine constraints. The majority of classical SDP solvers are designed for the deterministic setting where problem data is readily available. In this setting, generalized conditional gradient methods (aka Frank-Wolfe-type methods) provide scalable solutions by leveraging the so-called linear minimization oracle instead of the projection onto the semidefinite cone. Most problems in machine learning and modern engineering applications, however, contain some degree of stochasticity. In this work, we propose the first conditional-gradient-type method for solving stochastic optimization problems under affine constraints. Our method guarantees $\mathcal{O}(k^{-1/3})$ convergence rate in expectation on the objective residual and $\mathcal{O}(k^{-5/12})$ on the feasibility gap.

1 Introduction

We focus on the following stochastic convex composite optimization template, which covers finite sum and online learning problems:

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad \mathbb{E}_\Omega f(x, \omega) + g(Ax) := F(x). \tag{P}$$

In this optimization template, we consider the following setting:
▷ $\mathcal{X} \subset \mathbb{R}^n$ is a convex and compact set,
▷ ω is a realization of the random variable Ω drawn from the distribution $\mathcal{P}$,
▷ $\mathbb{E}_\Omega f(\cdot, \omega) : \mathcal{X} \to \mathbb{R}$ is a smooth (see Section 1.2 for the definition) convex function,
▷ $A \in \mathbb{R}^n \to R^d$ is a given linear map,
▷ $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a convex function (possibly non-smooth).

We consider two distinct specific cases for g:
(i) g is a Lipschitz-continuous function, for which the proximal-operator is easy to compute:

$$\text{prox}_g(y) = \arg\min_{z \in \mathbb{R}^d} \ g(z) + \frac{1}{2}\|z - y\|^2 \tag{1}$$

(ii) g is the indicator function of a convex set $\mathcal{K} \subset \mathbb{R}^d$:

$$g(z) = \begin{cases} 0 & \text{if } z \in \mathcal{K}, \\ +\infty & \text{otherwise}. \end{cases} \tag{2}$$

The former covers the regularized optimization problems. This type of regularization is common in machine learning applications to promote a desired structure to the solution. The latter handles affine constraints of the form $Ax \in \mathcal{K}$. We can also attack the combination of both: the minimization of a regularized loss-function subject to some affine constraints.

In this paper, we propose a conditional-gradient-type method (*aka* Frank-Wolfe-type) for (P). In summary, our main contributions are as follows:

▷ We propose the first CGM variant for solving (P). By CGM variant, we mean that our method avoids projection onto $\mathcal{X}$ and uses the *lmo* of $\mathcal{X}$ instead. The majority of the known methods for (P) require projections onto $\mathcal{X}$.

▷ We prove $\mathcal{O}(k^{-1/3})$ convergence rate on objective residual when g is Lipschitz-continuous.

▷ We prove $\mathcal{O}(k^{-1/3})$ convergence rate on objective residual, and $\mathcal{O}(k^{-5/12})$ on feasibility gap when g is an indicator function. Surprisingly, affine constraints that make the *lmo* challenging for existing CGM variants can be easily incorporated in this framework by using smoothing.

▷ We provide empirical evidence that validates our theoretical findings. Our results highlight the benefits of our framework against the projection-based algorithms.

1.1 Motivation: Stochastic Semidefinite Programming

Consider the following stochastic semidefinite programming template, minimization of a convex function over the positive-semidefinite cone subject to some affine constraints:

$$\underset{X \in \mathbb{S}_+^n, \ \mathrm{tr}(X) \leq \beta}{\text{minimize}} \quad \mathbb{E}_\Omega f(X, \omega) \quad \text{subject to} \quad AX \in \mathcal{K}. \tag{3}$$

Here, $\mathbb{S}_+^n$ denotes the positive-semidefinite cone. We are interested in solving (3) rather than the classical SDP since it does not require access to the whole data at one time. This creates a new vein of SDP applications in machine learning. Examples span online variants of clustering [33], streaming PCA [4], kernel learning [24], community detection [1], optimal power-flow [29], etc.

Example: Clustering. Consider the SDP formulation of the k-means clustering problem [33]:

$$\underset{X \in \mathbb{S}_+^n, \ \mathrm{tr}(X) = k}{\text{minimize}} \quad \langle D, \, X \rangle \quad \text{subject to} \quad X 1_n = 1_n, \quad X \geq 0. \tag{4}$$

Here, 1_n denotes the vector of ones, $X \geq 0$ enforces entrywise non-negativity, and D is the Euclidean distance matrix. Classical SDP solvers assume that we can access to the whole data matrix D at each time instance. By considering (3), we can solve this problem using only a subset of entries of D at each iteration. Remark that a subset of entries of D can be computed form a subset of the datapoints, since D is the Euclidean distance matrix.

We can attack (P), and (3) as a special case, by using operator splitting methods, assuming that we can efficiently project a point onto $\mathcal{X}$ (see [2] and the references therein). However, projection onto semidefinite cone might require a full eigendecomposition, which imposes a computational bottleneck (with its cubic cost) even for medium scaled problems with a few thousand dimensions.

When affine constraints are absent from the formulation (3), we can use stochastic CGM variants from the literature. The main workhorse of these methods is the so-called linear minimization oracle:

$$S = \arg\min_{Y} \ \left\{ \langle \nabla f(X, \omega), \, Y \rangle : \quad Y \in \mathbb{S}_+^n, \ \mathrm{tr}(Y) \leq \beta \right\} \tag{lmo}$$

We can compute S if we can find an eigenvector that corresponds to the smallest eigenvalue of $\nabla f(X, \omega)$. We can compute these eigenvectors efficiently by using shifted power methods or the randomized subspace iterations [12]. When we also consider affine constraints in our problem template, however, *lmo* becomes an SDP instance in the canonical form. In this setting, neither projection nor *lmo* is easy to compute. To our knowledge, no existent CGM variant is effective for solving (3) (and (P)). We specifically bridge this gap.

1.2 Notation and Preliminaries

We denote the expectation with respect to the random variable Ω by $\mathbb{E}_\Omega$, and the expectation wrt the sources of randomness in the optimization simply by $\mathbb{E}$. Furthermore we denote $f^\star := \mathbb{E}_\Omega f(x^\star, \omega)$ where $x^\star$ is the solution of (P). Throughout the paper, $y^\star$ represents the solution of the dual problem of (P). We assume that strong duality holds. Slater's condition is a common sufficient condition for strong duality that implies existence of a solution of the dual problem with finite norm.

Solution. We denote a solution to (P) and the optimal value by $x^\star$ and $F^\star$ respectively:

$$F^\star = F(x^\star) \le F(x), \qquad \forall x \in \mathcal{X}. \tag{5}$$

We say $x_\epsilon^\star \in \mathcal{X}$ is an ϵ-suboptimal solution (or simply an ϵ-solution) if and only if

$$F(x_\epsilon^\star) - F^\star \le \epsilon. \tag{6}$$

Stochastic first-order oracle (sfo). For the stochastic function $\mathbb{E}_\Omega f(x, \omega)$, suppose that we have access to a stochastic first-order oracle that returns a pair $(f(x, \omega), \nabla f(x, \omega))$ given x, where ω is an *iid* sample from distribution $\mathcal{P}$.

Lipschitz continuity & Smoothness. A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz continuous if

$$|g(z^1) - g(z^2)| \le L\|z^1 - z^2\|, \qquad \forall z^1, z^2 \in \mathbb{R}^d. \tag{7}$$

A differentiable function f is said to be L-smooth if the gradient ∇f is L-Lipschitz continuous.

2 Stochastic Homotopy CGM

Most stochastic CGM variants require mini-batch size to increase, in order to reduce the variance of the gradient estimator. However, Mokhtari et al., [31] have recently shown that the following (biased) estimator (that can be implemented with a single sample) can be incorporated with the CGM analysis:

$$d_k = (1 - \rho_k)d_{k-1} + \rho_k \nabla_x f(x_k, \omega_k) \tag{8}$$

The resulting method guarantees $\mathcal{O}(1/k^{\frac{1}{3}})$ convergence rate for convex smooth minimization, but it does not apply to our composite problem template (P).

Algorithm 1 SHCGM

Input: $x_1 \in \mathcal{X}$, $\beta_0 > 0$, $d_0 = 0$
for $k = 1, 2, \ldots,$ **do**
  $\eta_k = 9/(k + 8)$
  $\beta_k = \beta_0/(k + 8)^{\frac{1}{2}}$
  $\rho_k = 4/(k + 7)^{\frac{2}{3}}$
  $d_k = (1 - \rho_k)d_{k-1} + \rho_k \nabla_x f(x_k, \omega_k)$
  $v_k = d_k + \beta_k^{-1} A^\top \left(Ax_k - \mathrm{prox}_{\beta_k g}(Ax_k) \right)$
  $s_k = \arg\min_{x \in \mathcal{X}} \langle v_k, x \rangle$
  $x_{k+1} = x_k + \eta_k(s_k - x_k)$
end for

On the other hand, we introduced a CGM variant for composite problems (also covers affine constraints) in the deterministic setting in our prior work [41]. Our framework combines Nesterov smoothing [32] (and the quadratic penalty for affine constraints) with the CGM analysis. Unfortunately, this method does not work for stochastic problems.

In this paper, we propose the Stochastic Homotopy Conditional Gradient Method (SHCGM) for solving (P). The proposed method combines the stochastic CGM of [31] with our (deterministic) CGM for composite problems [41] in a non-trivial way.

Remark that the following formulation uniformly covers the Nesterov smoothing (with the Euclidean prox-function $\frac{1}{2}\|\cdot\|^2$) and the quadratic penalty (but the analyses for these two cases differ):

$$g_\beta(z) = \max_{y \in \mathbb{R}^d} \langle z, y \rangle - g^*(y) - \frac{\beta}{2}\|y\|^2, \quad \text{where} \quad g^*(x) = \max_{v \in \mathbb{R}^d} \langle x, v \rangle - g(v). \tag{9}$$

We call g_β as the smooth approximation of g, parametrized by the penalty (or smoothing) parameter $\beta > 0$. It is easy to show that g_β is $1/\beta$-smooth. Remark that the gradient of g_β can be computed by the following formula:

$$\nabla_x g_\beta(Ax) = A^\top \mathrm{prox}_{\beta^{-1}g^*}(\beta^{-1}Ax) = \beta^{-1} A^\top \left(Ax - \mathrm{prox}_{\beta g}(Ax) \right), \tag{10}$$

where the second equality follows from the Moreau decomposition.

The main idea is to replace the non-smooth component g by the smooth approximation g_β in (P). Clearly the solutions for (P) with $g(Ax)$ and $g_\beta(Ax)$ do not coincide for any value of β. However, $g_\beta \to g$ as $\beta \to 0$. Hence, we adopt a homotopy technique: We decrease β at a controlled rate as we progress in the optimization procedure, so that the decision variable converges to a solution of the original problem.

SHCGM is characterized by the following iterative steps:
▷ Decrease the step-size, smoothing and gradient averaging parameters η_k, β_k and ρ_k.
▷ Call the stochastic first-order oracle and compute the gradient estimator d_k in (8).
▷ Compute the gradient estimator v_k for the smooth approximation of the composite objective,

$$F_{\beta_k}(x) = \mathbb{E}_\Omega f(x, \omega) + g_{\beta_k}(Ax) \quad \implies \quad v_k = d_k + \nabla_x g_{\beta_k}(Ax). \tag{11}$$

▷ Compute the *lmo* with respect to v_k.
▷ Perform a CGM step to find the next iterate.

The roles of ρ_k and β_k are coupled. The former controls the variance of the gradient estimator, and the latter decides how fast we reduce the smoothing parameter to approach to the original problem. A carefully tuned interaction between these two parameters allows us to prove the following convergence rates.

Assumption (Bounded variance). We assume the following bounded variance condition holds:

$$\mathbb{E}\left[\|\nabla_x f(x, \omega) - \nabla_x \mathbb{E}_\Omega f(x, \omega)\|^2\right] \le \sigma^2 < +\infty. \tag{12}$$

Theorem 1 (Lipschitz-continuous regularizer). *Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 1 satisfies the following convergence bound:*

$$\mathbb{E}F(x_{k+1}) - F^\star \le 9^{\frac{1}{3}} \frac{C}{(k+8)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}, \tag{13}$$

where $C := \frac{81}{2} D_\mathcal{X}^2 (L_f + \beta_0 \|A\|^2) + 36\sigma D_\mathcal{X} + 27\sqrt{3} L_f D_\mathcal{X}^2$.

Proof sketch. The proof follows the following steps:
(i) Relate the stochastic gradient to the full gradient (Lemma 7).
(ii) Show convergence of the gradient estimator to the full gradient (Lemma 8).
(iii) Show $\mathcal{O}(1/k^{\frac{1}{3}})$ convergence rate on the smooth gap $\mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star$ (Theorem 9).
(iv) Translate this bound to the actual sub-optimality $\mathbb{E}F(x_{k+1}) - F^\star$ by using the envelope property for Nesterov smoothing, see Equation (2.7) in [32]. □

Convergence rate guarantees for stochastic CGM with Lipschitz continuous g (also based on Nesterov smoothing) are already known in the literature, see [16, 22, 23] for examples. Our rate is not faster than the ones in [22, 23], but we obtain $\mathcal{O}(\frac{1}{\epsilon^3})$ sample complexity in the statistical setting as opposed to $\mathcal{O}(\frac{1}{\epsilon^4})$.

In contrast with the existing stochastic CGM variants, our algorithm can also handle affine constraints. Remark that the indicator functions are not Lipschitz continuous, hence the Nesterov smoothing technique does not work for affine constraints.

Assumption (Strong duality). For problems with affine constraints, we further assume that the strong duality holds. Slater's condition is a common sufficient condition for strong duality. By Slater's condition, we mean

$$\text{relint}(\mathcal{X} \times \mathcal{K}) \cap \{(x, r) \in \mathbb{R}^n \times \mathbb{R}^d : Ax = r\} \ne \emptyset. \tag{14}$$

Recall that the strong duality ensures the existence of a finite dual solution.

Theorem 2 (Affine constraints). *Suppose that $g : \mathbb{R}^d \to \mathbb{R}$ is the indicator function of a simple convex set $\mathcal{K}$. Assuming that the strong duality holds, the sequence x_k generated by SHCGM satisfies*

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \ge -\|y^\star\| \, \mathbb{E}\text{dist}(Ax_{k+1}, \mathcal{K})$$

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \le 9^{\frac{1}{3}} \frac{C}{(k+8)^{\frac{1}{3}}} \tag{15}$$

$$\mathbb{E}\text{dist}(Ax_{k+1}, \mathcal{K}) \le \frac{2\beta_0 \|y^\star\|}{\sqrt{k+8}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}} C \beta_0}}{(k+8)^{\frac{5}{12}}}$$

Proof sketch. We re-use the ingredients of the proof of Theorem 1, except that at step (iv) we translate the bound on the smooth gap (penalized objective) to the actual convergence measures (objective residual and feasibility gap) by using the Lagrange saddle point formulations and the strong duality. See Corollaries 1 and 2. $\qquad\square$

Remark (Comparison to baseline). SHCGM combines ideas from [31] and [41]. Surprisingly,
$\triangleright \mathcal{O}(1/k^{\frac{1}{3}})$ rate in objective residual matches the rate in [31] for smooth minimization.
$\triangleright \mathcal{O}(1/k^{\frac{5}{12}})$ rate in feasibility gap is only an order of $k^{\frac{1}{12}}$ worse than the deterministic variant in [41].

Remark (Inexact oracles). We assume to use the exact solutions of *lmo* in SHCGM in Theorems 1 and 2. In many applications, however, it is much easier to find an approximate solution of *lmo*. For instance, this is the case for the SDP problems in Section 1.1. To this end, we extend our results for inexact *lmo* calls with additive and multiplicative error in the supplements.

Remark (Splitting). An important use-case of affine constraints in (P) is splitting (see Section 5.6 in [41]). Suppose that $\mathcal{X}$ can be written as the intersection of two (or more) simpler (in terms of computational cost of *lmo* or projection) sets $\mathcal{A} \cap \mathcal{B}$. By using the standard product space technique, we can reformulate this problem in the extended space $(x, y) \in \mathcal{A} \times \mathcal{B}$ with the constraint $x = y$:

$$\underset{(x,y) \in \mathcal{A} \times \mathcal{B}}{\text{minimize}} \quad \mathbb{E}_{\Omega} f(x, \omega) \quad \text{subject to} \quad x = y. \tag{16}$$

This allows us to decompose the difficult optimization domain $\mathcal{X}$ into simpler pieces. SHCGM requires *lmo* of $\mathcal{A}$ and *lmo* $\mathcal{B}$ separately. Alternatively, we can also use the projection onto one of the component sets (say $\mathcal{B}$) by reformulating the problem in domain $\mathcal{A}$ with an affine constraint $x \in \mathcal{B}$:

$$\underset{x \in \mathcal{A}}{\text{minimize}} \quad \mathbb{E}_{\Omega} f(x, \omega) \quad \text{subject to} \quad x \in \mathcal{B}. \tag{17}$$

An important example is the completely positive cone (intersection of the positive-semidefinite cone and the first orthant). Remark that the Clustering SDP example in Section 1.1 is also defined on this cone. While the *lmo* of this intersection can only be evaluated in $\mathcal{O}(n^3)$ computation by using the Hungarian method, we can compute the *lmo* for the semidefinite cone and the projection onto the first orthant much more efficiently.

3 Related Works

CGM dates back to the 1956 paper of Frank and Wolfe [8]. It did not acquire much interest in machine learning until the last decade because of its slower convergence rate in comparison with the (projected) accelerated gradient methods. However, there has been a resurgence of interest in CGM and its variants, following the seminal papers of Hazan [14] and Jaggi [18]. They demonstrate that CGM might offer superior computational complexity than state-of-the-art methods in many large-scale optimization problems (that arise in machine learning) despite its slower convergence rate, thanks to its lower per-iteration cost.

The original method by Frank and Wolfe [8] was proposed for smooth convex minimization on polytopes. The analysis is extended for smooth convex minimization on simplex by Clarkson [3], spactrahedron by Hazan [14], and finally for arbitrary compact convex sets by Jaggi [18]. All these methods are restricted for smooth problems.

Lan [21] proposed a variant for non-smooth minimization based on the Nesterov smoothing technique. Lan and Zhou [23] also introduced the conditional gradient sliding method and extended it for the non-smooth minimization in a similar way. These methods, however, are not suitable for solving (P) because we let g to be an indicator function which is not smoothing friendly.

In a prior work [41], we introduced homotopy CGM (HCGM) for composite problems (also with affine constraints). HCGM combines the Nesterov smoothing and quadratic penalty techniques under the CGM framework. It has $\mathcal{O}(1/\varepsilon^2)$ iteration complexity. In a follow-up work [40], we extended this method from quadratic penalty to an augmented Lagrangian formulation for empirical benefits. Gidel et al., [10] also proposed an augmented Lagrangian CGM but the analysis and guarantees differ. We refer to the references in [40, 41] for other variants in this direction.

So far, we have focused on deterministic variants of CGM. The literature on stochastic variants are much younger. We can trace it back to the Hazan and Kale's projection-free methods for online

learning [16]. When g is a non-smooth but Lipschitz continuous function, their method returns an ε-solution in $\mathcal{O}(1/\varepsilon^4)$ iterations.

The standard extension of CGM to the stochastic setting gets $\mathcal{O}(1/\varepsilon)$ iteration complexity for smooth minimization, but with an increasing minibatch size. Overall, this method requires $\mathcal{O}(1/\varepsilon^3)$ sample complexity, see [17] for the details. More recently, Mokhtari et al., [31] proposed a new variant with $\mathcal{O}(1/\varepsilon^3)$ convergence rate, but the proposed method can work with a single sample at each iteration. Hazan and Luo [17] and Yurtsever et al., [42] incorporated various variance for further improvements. Goldfarb et al., [11] introduced two stochastic CGM variants, with away-steps and pairwise-steps. These methods enjoy linear convergence rate (however, the batchsize increases exponentially) but for strongly convex objectives and only in polytope domains. None of these stochastic CGM variants work for non-smooth (or composite) problems.

Non-smooth conditional gradient sliding by Lan and Zhou [23] also have extensions to the stochastic setting. There is also a lazy variant with further improvements by Lan et al., [22]. Note however, similar to their deterministic variants, these methods are based on the Nesterov smoothing and are not suitable for problems with affine constraints.

Garber and Kaplan [9] considers problem (P). They also propose a variance reduced algorithm, but this method indeed solves the smooth relaxation of (P) (see Definition 1 Section 4.1). Contrary to SHCGM, this method might not asymptotically converge to a solution of the original problem.

Lu and Freund [28] also studied a similar problem template. However, their method incorporates the non-smooth term into the linear minimization oracle. This is restrictive in practice because the non-smooth term can increase the cost of linear minimization. In particular, this is the case when g is an indicator function, such as in SDP problems. This is orthogonal to our scenario in which the affine constraints are processed by smoothing, not directly through *lmo*.

In recent years, CGM has also been extended for non-convex problems. These extensions are beyond the scope of this paper. We refer to Yu et al., [39] and Julien-Lacoste [19] for the non-convex extensions in the deterministic setting, and to Reddi et al., [34], Yurtsever et al., [42], and Shen et al. [37] in the stochastic setting.

To the best of our knowledge, SHCGM is the first CGM-type algorithm for solving (P) with cheap linear minimization oracles. Another popular approach for solving large-scale instances of (P) is the operator splitting. See [2] and the references therein for stochastic operator splitting methods. Unfortunately, these methods still require projection onto $\mathcal{X}$ at each iteration. This projection is arguably more expensive than the linear minimization. For instance, for solving (3), the projection has cubic cost (with respect to the problem dimension n) while the linear minimization can be efficiently solved using subspace iterations, as depicted in Table 1.

Algorithm	Iteration complexity	Sample complexity	Solves (3)	Per-iteration cost (for (3))
[41]	$\mathcal{O}(1/\varepsilon^2)$	N	Yes	$\Theta(N_\nabla/\delta)$
[9]	$\mathcal{O}(1/\varepsilon^2)$	$\mathcal{O}(1/\varepsilon^4)$	No	$\Theta(N_\nabla/\delta)$
[17]	$\mathcal{O}(1/\varepsilon)$	$\mathcal{O}(1/\varepsilon^3)$	No	$\Theta(N_\nabla/\delta)$
[28]	$\mathcal{O}(1/\varepsilon)$	$\mathcal{O}(1/\varepsilon^2)$	No	SDP
[15]	$\mathcal{O}(1/\varepsilon)$	N	No	$\Theta(N_\nabla/\delta)$
[2]*	—	—	Yes	$\Theta(n^3)$
SHCGM	$\mathcal{O}(1/\varepsilon^3)$	$\mathcal{O}(1/\varepsilon^3)$	Yes	$\Theta(N_\nabla/\delta)$

Table 1: Existing algorithms to tackle (3). N is the size of the dataset. n is the dimension of each datapoint. N_∇ is the number of non-zeros of the gradient. δ is the accuracy of the approximate *lmo*. The per-iteration cost of [28] is the cost of solving a SDP in the canonical form.
*[2] has $\mathcal{O}(1/\varepsilon^2)$ iteration and sample complexity when the objective function is strongly convex. This is not the case in our model problem, and [2] only has an asymptotic convergence guarantee.

4   Numerical Evidence

This section presents the empirical performance of the proposed method for the stochastic k-means clustering, covariance matrix estimation, and matrix completion problems. We performed the experiments in MATLAB R2018a using a computing system of $4\times$ Intel Xeon CPU E5-2630 v3@2.40GHz and 16 GB RAM. We include the code to reproduce the results in the supplements[1].

4.1   Stochastic k-means Clustering

We consider the SDP formulation (4) of the k-means clustering problem. The same problem is used in numerical experiments by Mixon et al. [30], and we design our experiment based on their problem setup[2] with a sample of 1000 datapoints from the MNIST data[3]. See [30] for details on the preprocessing.

We solve this problem with SHCGM and compare it against HCGM [41] as the baseline. HCGM is a deterministic algorithm hence it uses the full gradient. For SHCGM, we compute a gradient estimator by randomly sampling 100 datapoints at each iteration. Remark that this corresponds to observing approximately 1 percent of the entries of D.

We use $\beta_0 = 1$ for HCGM and $\beta_0 = 10$ for SHCGM. We set these values by tuning both methods by trying $\beta_0 = 0.01, 0.1, ..., 1000$. We display the results in Figure 1 where we denote a full pass over the entries of D as an epoch. Figure 1 demonstrates that SHCGM performs similar to HCGM although it uses less data.
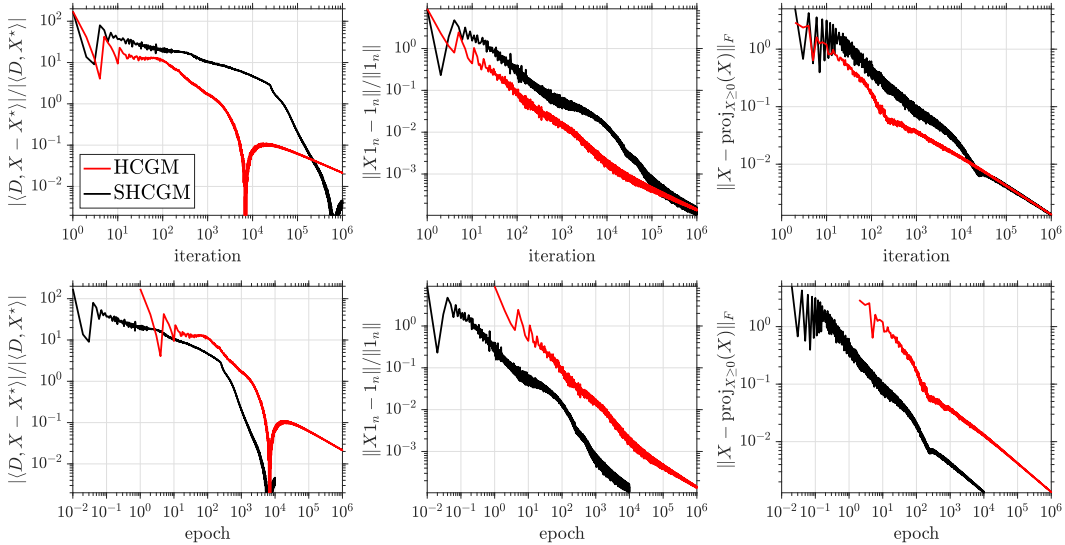

Figure 1: Comparison of SHCGM with HCGM for k-means clustering SDP in Section 4.1.

4.2   Online Covariance Matrix Estimation

Covariance matrix estimation is an important problem in multivariate statistics with applications in many fields including gene microarrays, social network, finance, climate analysis [35, 36, 7, 6], etc. In the online setting, we suppose that the data is received as a stream of datapoints in time.

The deterministic approach is to first collect some data, and then to train an empirical risk minimization model using the data collected. This has obvious limitations, since it may not be clear a priori how much data is enough to precisely estimate the covariance matrix. Furthermore, data can be too large to store or work with as a batch. To this end, we consider an online learning setting. In this case, we use each datapoint as it arrives and then discard it.

---

[1]Code also available at: `https://github.com/alpyurtsever/SHCGM`

[2]D.G. Mixon, S. Villar, R.Ward. — Available at `https://github.com/solevillar/kmeans_sdp`

[3]Y. LeCun and C. Cortes. — Available at `http://yann.lecun.com/exdb/mnist/`
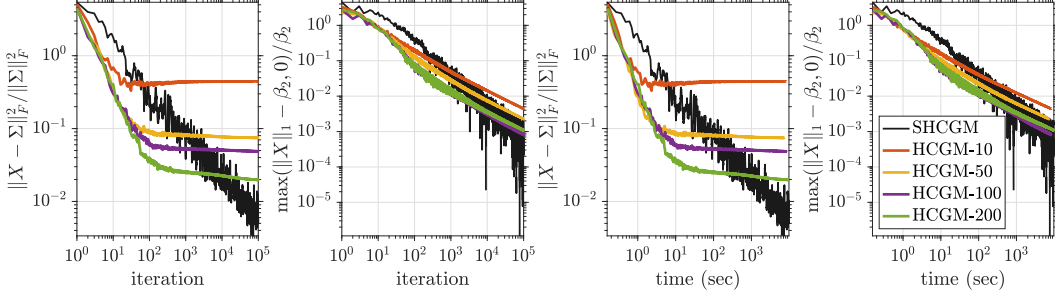
Figure 2: SHCGM and HCGM on Online covariance matrix estimation from streaming data.

Let us consider the following sparse covariance matrix estimation template (this template also covers other problems such as graph denoising and link prediction [35]) :

$$\underset{X \in \mathbb{S}_+^n, \ \operatorname{tr}(X) \le \beta_1}{\text{minimize}} \quad \mathbb{E}_\Omega \|X - \omega\omega^\top\|_F^2 \quad \text{subject to} \quad \|X\|_1 \le \beta_2. \tag{18}$$

where $\|X\|_1$ denotes the ℓ_1 norm (sum of absolute values of the entries).

Our test setup is as follows: We first create a block diagonal covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ using 10 blocks of the form $\phi\phi^\top$, where entries of ϕ are drawn uniformly random from $[-1, 1]$. This gives us a sparse matrix Σ of rank 10. Then, as for datapoints, we stream observations of Σ in the form $\omega_i \sim \mathcal{N}(0, \Sigma)$. We fix the problem dimension $n = 1000$.

We compare SHCGM with the deterministic method, HCGM. We use $\beta_0 = 1$ for both methods. Both methods require the *lmo* for the positive-semidefinite cone with trace constraint, and the projection oracle for the ℓ_1 norm constraint at each iteration.

We study two different setups: In Figure 2, we use SHCGM in the online setting. We sample a new datapoint at each iteration. HCGM, on the other hand, does not work in the online setting. Hence, we use the same sample of datapoints for all iterations. We consider 4 different cases with different sample sizes for HCGM, with 10, 50, 100 and 200 datapoints. Although this approach converges fast up to some accuracy, the objective value gets saturated at some estimation accuracy. Naturally, HCGM can achieve higher accuracy as the sample size increases.

We can also read the empirical convergence rates of SHCGM from Figure 2 as approximately $\mathcal{O}(k^{-1/2})$ for the objective residual and $\mathcal{O}(k^{-1})$ for the feasibility gap, significantly better than the theoretical guarantees .

If we can store larger samples, we can also consider minibatches for the stochastic methods. Figure 3 compares the deterministic approach with 200 datapoints with the stochastic approach with minibatch size of 200. In other words, while the deterministic method uses the same 200 datapoints for all iterations, we use a new draw of 200 datapoints at each iteration with SHCGM.
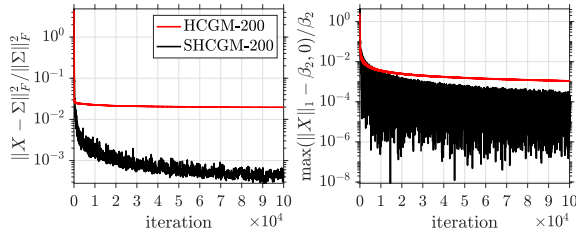

Figure 3: Comparison of SHCGM with HCGM batchsize 200 for online covariance matrix estimation.

4.3 Stochastic Matrix Completion

We consider the problem of matrix completion with the following mathematical formulation:

$$\underset{\|X\|_* \le \beta_1}{\text{minimize}} \quad \sum_{(i,j) \in \Omega} (X_{i,j} - Y_{i,j})^2 \quad \text{subject to} \quad 1 \le X \le 5, \tag{19}$$

where, Ω is the set of observed ratings (samples of entries from the true matrix Y that we try to recover), and $\|X\|_*$ denotes the nuclear-norm (sum of singular values). The affine constraint $1 \le X \le 5$ imposes a hard threshold on the estimated ratings (in other words, the entries of X).

8

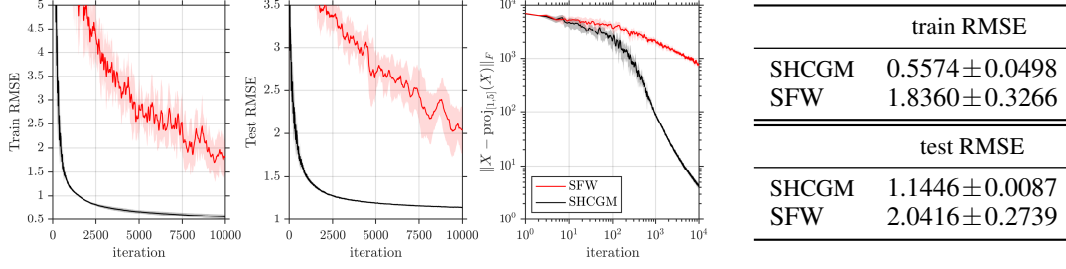|		train RMSE
	SHCGM	0.5574 ± 0.0498
	SFW	1.8360 ± 0.3266
		test RMSE
	SHCGM	1.1446 ± 0.0087
	SFW	2.0416 ± 0.2739

Figure 4: Training Error, Feasibility gap and Test Error for MovieLens 100k. Table shows the mean values and standard deviation of train and test RMSE over 5 different train/test splits at the end of 10^4 iterations.

We first compare SHCGM with the Stochastic Frank-Wolfe (SFW) from [31]. We consider a test setup with the MovieLens100k dataset[4] [13]. This dataset contains $\sim$100'000 integer valued ratings between 1 and 5, assigned by 1682 users to 943 movies. The aim of this experiment is to emphasize the flexibility of SHCGM: Recall that SFW does not directly apply to (19) as it cannot handle the affine constraint $1 \leq X \leq 5$. Therefore, we apply SFW to a relaxation of (19) that omits this constraint. Then, we solve (19) with SHCGM and compare the results.

We use the default `ub.train` and `ub.test` partitions provided with the original data. We set the model parameter for the nuclear norm constraint $\beta_1 = 7000$, and the initial smoothing parameter $\beta_0 = 10$. At each iteration, we compute a gradient estimator from 1000 *iid* samples. We perform the same test independently for 10 times to compute the average performance and confidence intervals. In Figure 4, we report the training and test errors (root mean squared error) as well as the feasibility gap. The solid lines display the average performance, and the shaded areas show $\pm$ one standard deviation. Note that SHCGM performs uniformly better than SFW, both in terms of the training and test errors. The Table shows the values achieved at the end of 10'000 iterations.

Finally, we compare SHCGM with the stochastic three-composite convex minimization method (S3CCM) from [43]. S3CCM is a projection-based method that applies to (19). In this experiment, we aim to demonstrate the advantages of the projection-free methods for problems in large-scale.

We consider a test setup with the MovieLens1m dataset[4] with $\sim$1 million ratings from $\sim$6000 users on $\sim$4000 movies. We partition the data into training and test samples with a $80/20$ train/test split. We use 10'000 *iid* samples at each iteration to compute a gradient estimator. We set the model parameter $\beta_1 = 20'000$. We use $\beta_0 = 10$ for SHCGM, and we set the step-size parameter $\gamma = 1$ for S3CCM. We implement the *lmo* efficiently using the power method. We refer to the code in the supplements for details on the implementation.

Figure 5 reports the outcomes of this experiment. SHCGM clearly outperforms S3CCM in this test. We run both methods for 2 hours. Within this time limit, SHCGM can perform 27'860 iterations while S3CCM can gets only up to 435 because of the high computational cost of the projection.
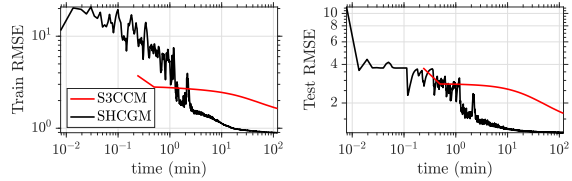

Figure 5: SHCGM vs S3CCM with MovieLens-1M.

5 Conclusions

We introduced a scalable stochastic CGM-type method for solving convex optimization problems with affine constraints and demonstrated empirical superiority of our approach in various numerical experiments. In particular, we consider the case of stochastic optimization of SDPs for which we give the first projection-free algorithm. In general, we showed that our algorithm provably converges to an optimal solution of (P) with $\mathcal{O}(k^{-1/3})$ and $\mathcal{O}(k^{-5/12})$ rates in the objective residual and feasibility gap respectively, with a sample complexity in the statistical setting of $\mathcal{O}(k^{-1/3})$. The possibility of a faster rate with the same (or even better) sample complexity remains an open question as well as an adaptive approach with $\mathcal{O}(k^{-1/2})$ rate when fed with exact gradients.

---

[4]F.M. Harper, J.A. Konstan. — Available at `https://grouplens.org/datasets/movielens/`

Acknowledgements

References

[1] E. Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18:1–86, 2018.

[2] V. Cevher, B. C. Vu, and A. Yurtsever. Stochastic forward Douglas-Rachford splitting method for monotone inclusions. In P. Giselsson and A. Rantzer, editors, *Large–Scale and Distributed Optimization*, chapter 7, pages 149–179. Springer International Publishing, 2018.

[3] K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4), 2010.

[4] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[5] C. Dünner, S. Forte, M. Takác, and M. Jaggi. Primal–dual rates and certificates. In *Proc. 33rd International Conference on Machine Learning*, 2016.

[6] J. Fan, F. Han, and H. Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

[7] J. Fan, Y. Liao, and H. Liu. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, 19(1):C1–C32, 2016.

[8] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[9] D. Garber and A. Kaplan. Fast stochastic algorithms for low-rank and nonsmooth matrix problems. *arXiv:1809.10477*, 2018.

[10] G. Gidel, F. Pedregosa, and S. Lacoste-Julien. Frank-Wolfe splitting via augmented Lagrangian method. In *Proc. 21st International Conference on Artificial Intelligence and Statistics*, 2018.

[11] D. Goldfarb, G. Iyengar, and C. Zhou. Linear convergence of stochastic Frank Wolfe variants. In *Proc. 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[12] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[13] F. M. Harper and J. A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[14] E. Hazan. Sparse approximate solutions to semidefinite programs. In *Proc. 8th Latin American Conf. Theoretical Informatics*, pages 306–316, 2008.

[15] E. Hazan. Sparse approximate solutions to semidefinite programs. In *Latin American symposium on theoretical informatics*, pages 306–316. Springer, 2008.

[16] E. Hazan and S. Kale. Projection–free online learning. In *Proc. 29th International Conference on Machine Learning*, 2012.

[17] E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *Proc. 33rd International Conference on Machine Learning*, 2016.

[18] M. Jaggi. Revisiting Frank–Wolfe: Projection–free sparse convex optimization. In *Proc. 30th International Conference on Machine Learning*, 2013.

[19] S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv:1607.00345*, 2016.

[20] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proc. 30th International Conference on Machine Learning*, 2013.

[21] G. Lan. The complexity of large–scale convex programming under a linear optimization oracle. *arXiv:1309.5550v2*, 2014.

[22] G. Lan, S. Pokutta, Y. Zhou, and D. Zink. Conditional accelerated lazy stochastic gradient descent. *arXiv:1703.05840*, 2017.

[23] G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM J. Optim.*, 26(2):1379–1409, 2016.

[24] G. R. G. Lanckriet, N. Cristianini, L. E. Ghaoui, P. Bartlett, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[25] F. Locatello, R. Khanna, M. Tschannen, and M. Jaggi. A unified optimization view on generalized matching pursuit and Frank-Wolfe. In *Proc. 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[26] F. Locatello, A. Raj, S. P. Karimireddy, G. Rätsch, B. Schölkopf, S. U. Stich, and M. Jaggi. On matching pursuit and coordinate descent. In *Proc. 35th International Conference on Machine Learning*, 2018.

[27] F. Locatello, M. Tschannen, G. Rätsch, and M. Jaggi. Greedy algorithms for cone constrained optimization with convergence guarantees. In *Advances in Neural Information Processing Systems 30*, 2017.

[28] H. Lu and R. M. Freund. Generalized stochastic frank-wolfe algorithm with stochastic" substitute"gradient for structured convex optimization. *arXiv:1807.07680*, 2018.

[29] J. L. R. Madani and S. Sojoudi. Convex relaxation for optimal power flow problem: mesh networks. *IEEE Trans. on Power Syst.*, 30(1):199–211, 2015.

[30] D. G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4):389–415, 2017.

[31] A. Mokhtari, H. Hassani, and A. Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv:1804.09554*, 2018.

[32] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, 2005.

[33] J. Peng and Y. Wei. Approximating K–means–type clustering via semidefinite programming. *SIAM J. Optim.*, 18(1):186–205, 2007.

[34] S. J. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic frank-wolfe methods for nonconvex optimization. *arXiv:1607.08254*, 2016.

[35] E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *Proc. 29th International Conference on Machine Learning*, 2012.

[36] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1), 2005.

[37] Z. Shen, C. Fang, P. Zhao, J. Huang, and H. Qian. Complexities in projection-free stochastic non-convex minimization. In *Proc. 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[38] Q. Tran-Dinh, O. Fercoq, and V. Cevher. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM J. Optim.*, 28(1):96–134, 2018.

[39] Y. Yu, X. Zhang, and D. Schuurmans. Generalized conditional gradient for sparse estimation. *arXiv:1410.4828v1*, 2014.

[40] A. Yurtsever, O. Fercoq, and V. Cevher. A conditional-gradient-based augmented Lagrangian framework. In *Proc. 36th International Conference on Machine Learning*, 2019.

[41] A. Yurtsever, O. Fercoq, F. Locatello, and V. Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *Proc. 35th International Conference on Machine Learning*, 2018.

[42] A. Yurtsever, S. Sra, and V. Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *Proc. 36th International Conference on Machine Learning*, 2019.

[43] A. Yurtsever, B. C. Vu, and V. Cevher. Stochastic three-composite convex minimization. In *Advances in Neural Information Processing Systems 29*, 2016.

A A Review of Smoothing

The technique described in [32] consists in a the following smooth approximation of a Lipschitz continuous function g as:

$$g_\beta(z) = \max_{y \in \mathbb{R}^d} \langle z, y \rangle - g^*(y) - \frac{\beta}{2} \|y\|^2,$$

where $\beta > 0$ controls the tightness of smoothing and g^* denotes the Fenchel conjugate of g

$$g^*(x) = \sup_{v \in \text{dom } g} \langle x, v \rangle - g(v).$$

It is easy to see that g_β is convex and $\frac{1}{\beta}$ smooth. Optimizing $g_\beta(z)$ guarantees progress on $g(z)$ when $g(z)$ is L_g-Lipschitz continuous as:

$$g_\beta(z) \leq g(z) \leq g_\beta(z) + \frac{\beta}{2} L_g$$

The challenge of smoothing an affine constraint consists in the fact that the indicator function is not Lipschitz. Therefore, g^* does not have bounded support so adding a strongly convex term to it does not guarantee that g and its smoothed version are uniformly close.

In order to smooth constraints which are not Nesterov smoothable, [41] consider an Homotopy transformation on β which can be intuitively understood as follows. If β decreases during the optimization, optimizing $g_\beta(z)$ will progressively become similar to optimizing $g(z)$. Therefore, the iterate will converge to the feasibility set.

Using the Homotopy smoothing, the objective of Equation (P) is replaced by the following approximation:

$$\min_{x \in \mathcal{X}} F_\beta(x) := \mathbb{E}_\Omega f(x, \omega) + g_\beta(Ax). \tag{20}$$

Let $y_{\beta_k}^*$ be:

$$y_{\beta_k}^*(Ax) = \arg\max_{y \in \mathbb{R}^d} \langle Ax, y \rangle - g^*(y) - \frac{\beta_k}{2} \|y\|^2 = \text{prox}_{\beta_k^{-1} g^*}(\beta_k^{-1} Ax) = \frac{1}{\beta_k} \big(Ax - \text{prox}_{\beta_k g}(Ax) \big),$$

with the last equality due to the Moreau decomposition.

Note that often $\text{prox}_{\beta_k g}(Ax)$ is easy to compute (for example when $g(z)$ is an affine constraint) but the projection on $\mathcal{X}$ is not. For example, for the $AX = b$ constraint of (3), $y_{\beta_k}^*(AX) = AX - b$. Therefore, [41] suggests to follow the same iterative procedure of the CGM which queries a Linear Minimization Oracle (lmo) at each iteration:

$$\text{lmo}(\nabla F_{\beta_k}) := \arg\min_{x \in \mathcal{X}} \langle x, \nabla F_{\beta_k}(x_k) \rangle \tag{21}$$

Moreover, we can compute the gradient of F_{β_k} as long as $\text{prox}_{\beta_k g}(Ax)$ is easy to compute. Indeed:

$$\nabla F_{\beta_k}(x) = \nabla_x \mathbb{E}_\Omega f(x, \omega) + A^\top y_{\beta_k}^*(Ax). \tag{22}$$

The solution of the *lmo* is then combined to the current iterate with a convex combination, so that the next iterate is guaranteed to be a member of $\mathcal{X}$. In the deterministic setting this technique comes with a reduction in the rate from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/\sqrt{k})$.

B Inexact Oracles

In practice, finding an exact solution can be expensive, especially when it involves a matrix factorization. Therefore, algorithms which are robust against inexact oracles are crucial in practice.

Even when the penalty is not present, we are not aware of approximate oracle rates in the framework of [31]. Due to the accumulation of the stochastic gradient, the deterministic definitions of inexact oracle are applicable to the stochastic case with the non-smooth penalty [20, 25, 27].

B.1 Additive Error

At iteration k, for the given v_k, we assume that the approximate *lmo* returns an element $\tilde{s}_k \in \mathcal{X}$ such that:

$$\langle v_k,\, \tilde{s}_k \rangle \leq \langle v_k,\, s_k \rangle + \delta \frac{\eta_k}{2} D_{\mathcal{X}}^2 \left(L_f + \frac{\|A\|^2}{\beta_k} \right) \tag{23}$$

for some $\delta > 0$, where s_k is the exact *lmo* solution.

We now present the convergence guarantees of Algorithm 1 when the exact *lmo* is replaced with the approximate oracle with additive error.

Corollary 3. *Assume that g is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 1 for $k \geq 1$ with approximate lmo (23) satisfies:*

$$\mathbb{E}F(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{k+8}},$$

where $C_\delta := \frac{81}{2} D_{\mathcal{X}}^2 (L_f + \beta_0\|A\|^2)(1+\delta) + 9 D_{\mathcal{X}} \sqrt{Q}$. We can optimize β_0 from this bound if δ is known.

Corollary 4. *Assume that g is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 1 with the lmo (23) satisfies:*

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \geq -\|y^\star\| \, \mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K})$$

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}$$

$$\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) \leq \frac{2\beta_0\|y^\star\|}{\sqrt{k+8}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}} C_\delta \beta_0}}{(k+8)^{\frac{5}{12}}}$$

B.2 Multiplicative Error

The additive error requires the accuracy of *lmo* to increase as the algorithm progresses [18]. This is restrictive in practice as it forces to invest more and more effort in the solution of the *lmo* problem.

For this reason, multiplicative error is often preferred, even though it adds the quality of the *lmo* as a hyperparmeter [25, 26], which we consider:

$$\langle v_k,\, \tilde{s}_k - x_k \rangle \leq \delta \langle v_k,\, s_k - x_k \rangle \tag{24}$$

where $\delta \in (0, 1]$ and s_k is the exact *lmo* solution.

We now present the convergence guarantees of Algorithm 1 when the exact *lmo* is replaced with the approximate oracle with multiplicative error (24)

Corollary 5. *Assume that g is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 1 with the lmo (24), and modifying $\eta_k = \frac{9}{\delta(k-1)+9}$, $\beta_k = \frac{\beta_0}{\sqrt{\delta(k-1)+9}}$ and $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$ satisfies:*

$$\mathbb{E}F(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}} \frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1)+9)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{\delta(k-1)+9}},$$

We can optimize β_0 from this bound if δ is known.

Corollary 6. *Assume that g is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 1 with approximate lmo (24), and modifying $\eta_k = \frac{9}{\delta(k-1)+9}$, $\beta_k =$*

$\frac{\beta_0}{\sqrt{\delta(k-1)+9}}$ and $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$ satisfies:

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \geq -\|y^\star\| \, \mathbb{E}\text{dist}(Ax_{k+1}, \mathcal{K})$$

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \leq 9^{\frac{1}{3}} \frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1)+9)^{\frac{1}{3}}}$$

$$\mathbb{E}\text{dist}(Ax_{k+1}, \mathcal{K}) \leq \frac{2\beta_0 \|y^\star\|}{\sqrt{\delta(k-1)+9}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}}(\frac{C}{\delta} + \mathcal{E}_1)\beta_0}}{(\delta(k-1)+9)^{\frac{5}{12}}}$$

C   Convergence Rate

We first prove some key lemmas. This section builds on top of the analysis of [31] and the homotopy CGM framework. All these results are for the inexact oracle with additive error, the exact oracle case can be obtained setting $\delta = 0$.

Lemma 7. *For any given iteration $k \geq 1$ of Algorithm 1 the following relation holds:*

$$\langle \nabla F_{\beta_k}(x_k), s_k - x_k \rangle \leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_\mathcal{X} + f^\star - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k)$$

$$- \frac{\beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2 + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right)$$

where $\delta \geq 0$ is the accuracy of the inexact lmo with additive error.

Proof.

$$\langle \nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k \rangle = \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega), \tilde{s}_k - x_k \rangle + \langle A^\top \nabla g_{\beta_k}(Ax_k), \tilde{s}_k - x_k \rangle$$

$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega), \tilde{s}_k - x_k \rangle + \langle A^\top \nabla g_{\beta_k}(Ax_k), \tilde{s}_k - x_k \rangle$$
$$+ \langle d_k, \tilde{s}_k - x_k \rangle - \langle d_k, \tilde{s}_k - x_k \rangle$$

$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k, \tilde{s}_k - x_k \rangle + \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k), \tilde{s}_k - x_k \rangle$$

$$\leq \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k, s_k - x_k \rangle$$

$$+ \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k), s_k - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right) \tag{25}$$

$$\leq \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k, s_k - x_k \rangle$$

$$+ \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right) \tag{26}$$

$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k, s_k - x_k \rangle + \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle$$

$$+ \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega), x^\star - x_k \rangle - \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega), x^\star - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right)$$

$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k, s_k - x^\star \rangle$$

$$+ \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right)$$

$$\leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| \|s_k - x^\star\|$$

$$+ \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right) \tag{27}$$

$$\leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_\mathcal{X}$$

$$+ \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle + \delta \frac{\eta_k}{2} D_\mathcal{X}^2 \left(L_f + \frac{\|A\|^2}{\beta_k}\right) \tag{28}$$

where Equation (25) is the definition of inexact oracle with additive error, Equation (26) is because s_k is a solution of $\min_{x \in \mathcal{X}} \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k), x \rangle$, Equation (27) is Cauchy-Schwarz

and the Equation (28) is the definition of diameter. Now, convexity of $\mathbb{E}_\Omega f(x_k, \omega)$ ensures $\langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega), x^\star - x_k \rangle \leq f^\star - \mathbb{E}_\Omega f(x_k, \omega)$. From Lemma 10 in [38] we have that:

$$g(z_1) \geq g_\beta(z_2) + \langle \nabla g_\beta(z_2), z_1 - z_2 \rangle + \frac{\beta}{2} \|y_\beta^*(z_2)\|^2. \tag{29}$$

Therefore:

$$\begin{aligned}
\langle A^\top \nabla g_{\beta_k}(Ax_k), x^\star - x_k \rangle &= \langle \nabla g_{\beta_k}(Ax_k), Ax^\star - Ax_k \rangle \\
&\leq g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2} \|y_{\beta_k}^*(Ax_k)\|^2.
\end{aligned}$$

Therefore:

$$\begin{aligned}
\langle \nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k \rangle &\leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}} + f^\star) - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k) \\
&\quad - \frac{\beta_k}{2} \|y_{\beta_k}^*(Ax_k)\|^2 + \delta \frac{\eta_k}{2} D_{\mathcal{X}}^2 \left(L_f + \frac{\|A\|^2}{\beta_k} \right)
\end{aligned}$$

$\square$

Lemma 8. *For any $k \geq 1$ the estimate of the gradient computed in Algorithm 1 satisfies:*

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2 \right] \leq \frac{Q}{(k+8)^{\frac{2}{3}}}$$

where $Q = \max \left\{ \|\nabla_x \mathbb{E}_\Omega f(x_1, \omega) - d_1\|^2 7^{\frac{2}{3}}, 16\sigma^2 + 81 L_f^2 D_{\mathcal{X}}^2 \right\}$

Proof. This lemma simply applies Lemma 1 and Lemma 17 of [31] to our different stepsizes. We report all the steps for clarity and completeness. First, we invoke Lemma 1:

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2 \right] \leq \left(1 - \frac{\rho_k}{2} \right) \|\nabla_x \mathbb{E}_\Omega f(x_{k-1}, \omega) - d_{k-1}\|^2 + \rho_k^2 \sigma^2 + \frac{2 L_f^2 D_{\mathcal{X}}^2 \eta_{k-1}^2}{\rho_k}$$

$$\leq \left(1 - \frac{2}{(k+7)^{\frac{2}{3}}} \right) \|\nabla_x \mathbb{E}_\Omega f(x_{k-1}, \omega) - d_{k-1}\|^2 + \frac{16\sigma^2 + 81 L_f^2 D_{\mathcal{X}}^2}{(k+7)^{\frac{4}{3}}} \tag{30}$$

where we used $\rho_k = \frac{4}{(k+7)^{\frac{2}{3}}}$. Now, Lemma 17 of [31] gives the following solution:

$$\phi_t \leq \frac{Q}{(k + k_0 + 1)^\alpha}$$

to the recursion

$$\phi_k \leq \left(1 - \frac{c}{(k + k_0)^\alpha} \right) \phi_{k-1} + \frac{b}{(k + k_0)^{2\alpha}}$$

where $b \geq 0$, $c > 1$, $\alpha \leq 1$, $k_0 \geq 0$ and $Q := \max \{ \phi_1 k_0^\alpha, b/(c-1) \}$ Applying this lemma to Equation (30) with $k_0 = 7$, $\alpha = \frac{2}{3}$, $c = 2$, $b = 16\sigma^2 + 81 L_f^2 D_{\mathcal{X}}^2$ gives:

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2 \right] \leq \frac{Q}{(k+8)^{\frac{2}{3}}}$$

where $Q = \max \left\{ \|\nabla \mathbb{E}_\Omega f(x_1, \omega) - d_1\|^2 7^{\frac{2}{3}}, 16\sigma^2 + 81 L_f^2 D_{\mathcal{X}}^2 \right\}$ $\square$

C.1 Proof of Theorem 9

We prove Theorem 9 with the oracle with additive error. The proof without additive error can be obtained with $\delta = 0$.

Theorem 9. *The sequence x_k generated by Algorithm 1 satisfies the following bound for $k \geq 1$:*

$$\mathbb{E} F_{\beta_k}(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}},$$

where $C_\delta := \frac{81}{2} D_{\mathcal{X}}^2 (L_f + \beta_0 \|A\|^2)(1 + \delta) + 9 D_{\mathcal{X}} \sqrt{Q}$, $Q = \max \left\{ 4\|\nabla \mathbb{E}_\Omega f(x_1, \omega) - d_1\|^2, 16\sigma^2 + 2 L_f^2 D_{\mathcal{X}}^2 \right\}$ and $\delta \geq 0$.

Proof. Note that Theorem 9 can be obtained as a special case setting $\delta = 0$. First, we use the smoothness of F_{β_k} to upper bound the progress. Note that F_{β_k} is $(L_f + \|A\|^2/\beta_k)$-smooth.

$$F_{\beta_k}(x_{k+1}) \leq F_{\beta_k}(x_k) + \eta_k \langle \nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k \rangle + \frac{\eta_k^2}{2} \|\tilde{s}_k - x_k\|^2 (L_f + \frac{\|A\|^2}{\beta_k})$$

$$\leq F_{\beta_k}(x_k) + \eta_k \langle \nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k \rangle + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k}), \quad (31)$$

where s_k denotes the atom selected by the *lmo*, and the second inequality follows since $s_k \in \mathcal{X}$. We now apply Lemma 7 and obtain:

$$F_{\beta_k}(x_{k+1}) \leq F_{\beta_k}(x_k) + \eta_k \left(f^\star - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2} \|\nabla y_{\beta_k}^*(Ax_k)\|^2 \right)$$

$$+ \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta) + \eta_k \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}} \quad (32)$$

$$= (1 - \eta_k) F_{\beta_k}(x_k) + \eta_k F^\star - \frac{\eta_k \beta_k}{2} \|\nabla y_{\beta_k}^*(Ax_k)\|^2 + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \eta_k \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}.$$

Now, using Lemma 10 of [38] we get:

$$g_\beta(z_1) \leq g_\gamma(z_1) + \frac{\gamma - \beta}{2} \|y_\beta^*(z_1)\|^2 \quad (33)$$

and therefore:

$$F_{\beta_k}(x_k) = \mathbb{E}_\Omega f(x_k, \omega) + g_{\beta_k}(Ax_k)$$

$$\leq \mathbb{E}_\Omega f(x_k, \omega) + g_{\beta_{k-1}}(Ax_k) + \frac{\beta_{k-1} - \beta_k}{2} \|y_{\beta_k}^*(Ax_k)\|^2$$

$$= F_{\beta_{k-1}}(x_k) + \frac{\beta_{k-1} - \beta_k}{2} \|y_{\beta_k}^*(Ax_k)\|^2.$$

We combine this with (32) and subtract $F^\star$ from both sides to get

$$F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \eta_k)\big(F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \big((1 - \eta_k)(\beta_{k-1} - \beta_k) - \eta_k \beta_k\big)\frac{1}{2} \|y_{\beta_k}^*(Ax_k)\|^2 + \eta_k \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}.$$

Let us choose η_k and β_k in a way to vanish the last term. By choosing $\eta_k = \frac{9}{k+8}$ and $\beta_k = \frac{\beta_0}{(k+8)^{\frac{1}{2}}}$ for $k \geq 1$ with some $\beta_0 > 0$, we get $(1 - \eta_k)(\beta_{k-1} - \beta_k) - \eta_k \beta_k < 0$. Hence, we end up with

$$F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \eta_k)\big(F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \eta_k \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}.$$

We now compute the expectation, use Jensen inequality and use Lemma 8 to obtain the final recursion:

$$\mathbb{E} F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \eta_k)\big(\mathbb{E} F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \eta_k \mathbb{E} \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}$$

$$\leq (1 - \eta_k)\big(\mathbb{E} F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \eta_k \sqrt{\mathbb{E} \|\nabla \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2} D_{\mathcal{X}}$$

$$\leq (1 - \eta_k)\big(\mathbb{E} F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1 + \delta)$$

$$+ \frac{9 D_{\mathcal{X}} \sqrt{Q}}{(k+8)^{\frac{4}{3}}}$$

Now, note that:

$$\frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k}) = \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 L_f + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 \frac{\|A\|^2}{\beta_k}$$

$$= \frac{\frac{81}{2}}{(k+8)^2} D_{\mathcal{X}}^2 L_f + \frac{\frac{81}{2}}{(k+8)^{\frac{3}{2}}} \beta_0 D_{\mathcal{X}}^2 \|A\|^2$$

$$\leq \frac{\frac{81}{2}}{(k+8)^{\frac{4}{3}}} D_{\mathcal{X}}^2 L_f + \frac{\frac{81}{2}}{(k+8)^{\frac{4}{3}}} \beta_0 D_{\mathcal{X}}^2 \|A\|^2$$

Therefore:

$$\mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star \leq \left(1 - \frac{9}{k+8}\right) \left(\mathbb{E}F_{\beta_{k-1}}(x_k) - F^\star\right)$$

$$+ \frac{\frac{81}{2} D_{\mathcal{X}}^2 (L_f + \beta_0 \|A\|^2)(1+\delta) + 9D_{\mathcal{X}}\sqrt{Q}}{(k+8)^{\frac{4}{3}}}$$

For simplicity, let $C_\delta := \frac{81}{2} D_{\mathcal{X}}^2 (L_f + \beta_0 \|A\|^2)(1+\delta) + 9D_{\mathcal{X}}\sqrt{Q}$ and $\mathcal{E}_{k+1} := \mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star$. Then, we need to solve the following recursive equation:

$$\mathcal{E}_{k+1} \leq \left(1 - \frac{9}{k+8}\right) \mathcal{E}_k + \frac{C_\delta}{(k+8)^{\frac{4}{3}}} \tag{34}$$

Let the induction hypothesis for $k \geq 1$ be:

$$\mathcal{E}_{k+1} \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}$$

For the base case $k = 1$ we need to prove $\mathcal{E}_2 \leq C_\delta$. From Equation (34) we have $\mathcal{E}_2 \leq \frac{C_\delta}{(9)^{\frac{4}{3}}} < C_\delta$ as $9^{\frac{4}{3}} > 1$ Now:

$$\mathcal{E}_{k+1} \leq \left(1 - \frac{9}{k+8}\right) \mathcal{E}_k + \frac{C_\delta}{(k+8)^{\frac{4}{3}}}$$

$$\leq \left(1 - \frac{9}{k+8}\right) 9^{\frac{1}{3}} \frac{C_\delta}{(k+7)^{\frac{1}{3}}} + \frac{C_\delta}{(k+8)^{\frac{4}{3}}}$$

$$\leq \left(1 - \frac{9}{k+8}\right) 9^{\frac{1}{3}} \frac{C_\delta}{(k+7)^{\frac{1}{3}}} + 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{4}{3}}}$$

$$\leq \left(1 - \frac{9}{k+8}\right) 9^{\frac{1}{3}} \frac{C_\delta}{(k+7)^{\frac{1}{3}}} + 9^{\frac{1}{3}} \frac{C_\delta}{(k+7)^{\frac{1}{3}}(k+8)}$$

$$= 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)(k+7)^{\frac{1}{3}}} (k-1)$$

$$\leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}$$

$\square$

Corollary' 1. *Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 1 satisfies the following convergence bound for $k \geq 1$:*

$$\mathbb{E}F(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}.$$

Proof. The proof is trivial and the technique comes from [41]. We report it for completeness. If $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is L_g-Lipschitz continuous from equation (2.7) in [32] and the duality between Lipshitzness and bounded support (*cf.* Lemma 5 in [5]) we have:

$$g_\beta(z) \leq g(z) \leq g_\beta(z) + \frac{\beta}{2} L_g^2 \tag{35}$$

17

Using this fact, we write:

$$g(Ax_{k+1}) \leq g_{\beta_k}(Ax_{k+1}) + \frac{\beta_k L_g^2}{2}$$

$$= g_{\beta_k}(Ax_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}.$$

We complete the proof by adding $\mathbb{EE}_\Omega f(x_{k+1}, \omega) - F^\star$ to both sides:

$$\mathbb{E}F(x_{k+1}) - F^\star \leq \mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}$$

$$\leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}.$$

$\square$

Corollary' 2. *Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 1 satisfies:*

$$\mathbb{EE}_\Omega f(x_k, \omega) - f^\star \geq -\|y^\star\| \, \mathbb{E}\mathrm{dist}(Ax_k, \mathcal{K})$$

$$\mathbb{EE}_\Omega f(x_k, \omega) - f^\star \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}$$

$$\mathbb{E}\mathrm{dist}(Ax_k, \mathcal{K}) \leq \frac{2\beta_0\|y^\star\|}{\sqrt{k+8}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}} C_\delta \beta_0}}{(k+8)^{\frac{5}{12}}}$$

Proof. We adapt to our rate the proof technique of Theorem 4.3 in [41]. From the Lagrange saddle point theory, we know that the following bound holds $\forall x \in \mathcal{X}$ and $\forall r \in \mathcal{K}$:

$$f^\star \leq \mathcal{L}(x, r, y^\star) = \mathbb{E}_\Omega f(x, \omega) + \langle y_\star, \, Ax - r \rangle$$

$$\leq \mathbb{E}_\Omega f(x, \omega) + \|y_\star\|\|Ax - r\|,$$

Since $x_{k+1} \in \mathcal{X}$ and taking the expectation, we get

$$\mathbb{EE}_\Omega f(x_{k+1}, \omega) - f^\star \geq -\mathbb{E}\min_{r \in \mathcal{K}} \|y^\star\|\|Ax_{k+1} - r\|$$

$$= -\|y^\star\|\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}). \qquad (36)$$

This proves the first bound in Corollary 2.

The second bound directly follows by Theorem 9 as

$$\mathbb{EE}_\Omega f(x_{k+1}, \omega) - f^\star \mathbb{E} \leq \mathbb{EE}_\Omega f(x_{k+1}, \omega) - f^\star + \frac{1}{2\beta_k}\mathbb{E}\left[\mathrm{dist}(Ax_{k+1}, \mathcal{K})\right]^2$$

$$\leq \mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star$$

$$\leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}.$$

Now, we combine this with (36), and we get

$$-\|y^\star\|\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) + \frac{1}{2\beta_k}\mathbb{E}\left[\mathrm{dist}(Ax_{k+1}, \mathcal{K})\right]^2 \leq 9^{\frac{1}{3}} \frac{C_\delta}{(k+8)^{\frac{1}{3}}}$$

This is a second order inequality in terms of $\mathbb{E}\mathrm{dist}(Ax_k, \mathcal{K})$. Solving this inequality, we get

$$\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) \leq \frac{2\beta_0\|y^\star\|}{\sqrt{k+8}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}} C_\delta \beta_0}}{(k+8)^{\frac{5}{12}}}.$$

$\square$

D Inexact Oracle with Multiplicative Error

Lemma 10. *For any given iteration $k \geq 1$ of Algorithm 1 the following relation holds:*

$$\langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k \rangle \leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}$$
$$+ \delta \left[f^\star - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2} \|y^*_{\beta_k}(Ax_k)\|^2 \right]$$

where $\delta \in (0, 1]$ is the accuracy of the inexact lmo with multiplicative error.

Proof.

$$\langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k \rangle = \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega),\, \tilde{s}_k - x_k \rangle + \langle A^\top \nabla g_{\beta_k}(Ax_k),\, \tilde{s}_k - x_k \rangle$$
$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega),\, \tilde{s}_k - x_k \rangle + \langle A^\top \nabla g_{\beta_k}(Ax_k),\, \tilde{s}_k - x_k \rangle + \langle d_k,\, \tilde{s}_k - x_k \rangle - \langle d_k,\, \tilde{s}_k - x_k \rangle$$
$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k,\, \tilde{s}_k - x_k \rangle + \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k),\, \tilde{s}_k - x_k \rangle$$
$$\leq \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k,\, \tilde{s}_k - x_k \rangle + \delta \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k),\, s_k - x_k \rangle \tag{37}$$
$$\leq \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k,\, \tilde{s}_k - x_k \rangle + \delta \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle \tag{38}$$
$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k,\, \tilde{s}_k - x_k \rangle + \delta \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle$$
$$+ \delta \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega),\, x^\star - x_k \rangle - \delta \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega),\, x^\star - x_k \rangle$$
$$= \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k,\, \tilde{s}_k - x_k - \delta x^\star + \delta x_k \rangle$$
$$+ \delta \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle$$
$$\leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| \|\tilde{s}_k - ((1 - \delta)x_k + \delta x^\star)\|$$
$$+ \delta \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle \tag{39}$$
$$\leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}} + \delta \langle \nabla_x \mathbb{E}_\Omega f(x_k, \omega) + A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle \tag{40}$$

where the Equation (37) is the definition of inexact oracle with multiplicative error, Equation (38) is because s_k is a solution of $\min_{x \in \mathcal{X}} \langle d_k + A^\top \nabla g_{\beta_k}(Ax_k),\, x \rangle$, Equation (39) is cauchy-schwarz and Equation (40) is the diameter definition noting that $(1 - \delta)x_k + \delta x^\star \in \mathcal{X}$ as it is a convex combination of elements in $\mathcal{X}$.

Now, convexity of $\mathbb{E}_\Omega f(x_k, \omega)$ ensures $\langle \nabla \mathbb{E}_\Omega f(x_k, \omega), x^\star - x_k \rangle \leq f^\star - \mathbb{E}_\Omega f(x_k, \omega)$. Using property (29), we have

$$\langle A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k \rangle = \langle \nabla g_{\beta_k}(Ax_k),\, Ax^\star - Ax_k \rangle$$
$$\leq g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2} \|y^*_{\beta_k}(Ax_k)\|^2.$$

Therefore:

$$\langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k \rangle \leq \|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\| D_{\mathcal{X}}$$
$$+ \delta \left[f^\star - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2} \|y^*_{\beta_k}(Ax_k)\|^2 \right]$$

$\square$

Lemma 11. *For any $k \geq 1$ the estimate of the gradient computed in Algorithm 1 satisfies:*

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2 \right] \leq \frac{Q}{(\delta(k - 1) + 9)^{\frac{2}{3}}}$$

where $Q = \max \left\{ \|\nabla_x \mathbb{E}_\Omega f(x_1, \omega) - d_1\|^2 7^{\frac{2}{3}},\, 16\sigma^2 + 81 L_f^2 D_{\mathcal{X}}^2 \right\}$

Proof. This lemma simply applies Lemma 1 and Lemma 17 of [31] to our different stepsizes. We report all the steps for clarity and completeness. First, we invoke Lemma 1:

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2\right] \leq \left(1 - \frac{\rho_k}{2}\right)\|\nabla_x \mathbb{E}_\Omega f(x_{k-1}, \omega) - d_{k-1}\|^2 + \rho_k^2 \sigma^2 + \frac{2L_f^2 D_\mathcal{X}^2 \eta_{k-1}^2}{\rho_k}$$

$$\leq \left(1 - \frac{2}{(\delta(k-2) + 9)^{\frac{2}{3}}}\right)\|\nabla_x \mathbb{E}_\Omega f(x_{k-1}, \omega) - d_{k-1}\|^2 + \frac{16\sigma^2 + 81L_f^2 D_\mathcal{X}^2}{(\delta(k-2) + 9)^{\frac{4}{3}}} \tag{41}$$

where we used $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$. Now, Lemma 17 of [31] gives the following solution:

$$\phi_t \leq \frac{Q}{(k + k_0 + 1)^\alpha}$$

to the recursion

$$\phi_k \leq \left(1 - \frac{c}{(k + k_0)^\alpha}\right)\phi_{k-1} + \frac{b}{(k + k_0)^{2\alpha}}$$

where $b \geq 0$, $c > 1$, $\alpha \leq 1$, $k_0 \geq 0$ and $\tilde{Q} := \max\{\phi_1 k_0^\alpha, b/(c-1)\}$ Applying this lemma to Equation (41) with $k_0 = \frac{9}{\delta} - 2$, $\alpha = \frac{2}{3}$, $c = \frac{2}{\delta^{\frac{2}{3}}}$, $b = \frac{16\sigma^2 + 81L_f^2 D_\mathcal{X}^2}{\delta^{\frac{4}{3}}}$ gives:

$$\mathbb{E}\left[\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2\right] \leq \frac{Q}{(\delta(k-1) + 9)^{\frac{2}{3}}}$$

where $Q = \max\left\{\|\nabla_x \mathbb{E}_\Omega f(x_1, \omega) - d_1\|^2(9 - 2\delta)^{\frac{2}{3}}, \frac{16\sigma^2 + 81L_f^2 D_\mathcal{X}^2}{(2 - \delta^{\frac{2}{3}})}\right\}$ □

Theorem 12. *The sequence x_k generated by Algorithm 1 with approximate lmo of the form (24), and modifying $\eta_k = \frac{9}{\delta(k-1)+9}$, $\beta_k = \frac{\beta_0}{\sqrt{\delta(k-1)+9}}$ and $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$ satisfies:*

$$\mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1) + 9)^{\frac{1}{3}}}$$

where $\mathcal{E}_1 := F_{\frac{\beta_0}{\sqrt{9}}}(x_1) - F^\star$.

Proof. First, we use the smoothness of F_{β_k} to upper bound the progress. Note that F_{β_k} is $(L_f + \|A\|^2/\beta_k)$-smooth.

$$F_{\beta_k}(x_{k+1}) \leq F_{\beta_k}(x_k) + \eta_k\langle\nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k\rangle + \frac{\eta_k^2}{2}\|\tilde{s}_k - x_k\|^2(L_f + \frac{\|A\|^2}{\beta_k})$$

$$\leq F_{\beta_k}(x_k) + \eta_k\langle\nabla F_{\beta_k}(x_k), \tilde{s}_k - x_k\rangle + \frac{\eta_k^2}{2}D_\mathcal{X}^2(L_f + \frac{\|A\|^2}{\beta_k}), \tag{42}$$

where $\tilde{s}_k$ denotes the atom selected by the approximate *lmo* with multiplicative accuracy, and the second inequality follows since $\tilde{s}_k \in \mathcal{X}$.

Using Lemma 10 we get:

$$F_{\beta_k}(x_{k+1}) \leq F_{\beta_k}(x_k) + \eta_k\delta\left(f^\star - \mathbb{E}_\Omega f(x_k, \omega) + g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2}\|\nabla y_{\beta_k}^*(Ax_k)\|^2\right)$$

$$+ \frac{\eta_k^2}{2}D_\mathcal{X}^2(L_f + \frac{\|A\|^2}{\beta_k}) + \eta_k\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|D_\mathcal{X} \tag{43}$$

$$= (1 - \delta\eta_k)F_{\beta_k}(x_k) + \delta\eta_k F^\star - \frac{\delta\eta_k\beta_k}{2}\|\nabla y_{\beta_k}^*(Ax_k)\|^2$$

$$+ \frac{\eta_k^2}{2}D_\mathcal{X}^2(L_f + \frac{\|A\|^2}{\beta_k}) + \eta_k\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|D_\mathcal{X}.$$

Now, using (33), we get

$$F_{\beta_k}(x_k) = \mathbb{E}_\Omega f(x_k, \omega) + g_{\beta_k}(Ax_k)$$

$$\leq \mathbb{E}_\Omega f(x_k, \omega) + g_{\beta_{k-1}}(Ax_k) + \frac{\beta_{k-1} - \beta_k}{2} \|y^*_{\beta_k}(Ax_k)\|^2$$

$$= F_{\beta_{k-1}}(x_k) + \frac{\beta_{k-1} - \beta_k}{2} \|y^*_{\beta_k}(Ax_k)\|^2.$$

We combine this with (43) and subtract $F^\star$ from both sides to get

$$F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \delta\eta_k)\big(F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big)$$

$$+ \big((1 - \delta\eta_k)(\beta_{k-1} - \beta_k) - \delta\eta_k\beta_k\big)\frac{1}{2}\|y^*_{\beta_k}(Ax_k)\|^2 + \eta_k\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|D_{\mathcal{X}}.$$

Let us choose η_k and β_k in a way to vanish the last term. By choosing $\eta_k = \frac{9}{\delta(k-1)+9}$ and $\beta_k = \frac{\beta_0}{(\delta(k-1)+9)^{\frac{1}{2}}}$ for $k \geq 1$ with some $\beta_0 > 0$, we get $(1 - \delta\eta_k)(\beta_{k-1} - \beta_k) - \delta\eta_k\beta_k < 0$. Hence, we end up with

$$F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \delta\eta_k)\big(F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big) + \eta_k\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|D_{\mathcal{X}}.$$

We now compute the expectation, use Jensen inequality and use Lemma 8 to obtain the final recursion:

$$\mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star \leq (1 - \delta\eta_k)\big(\mathbb{E}F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big)$$

$$+ \eta_k \mathbb{E}\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|D_{\mathcal{X}}$$

$$\leq (1 - \delta\eta_k)\big(\mathbb{E}F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big)$$

$$+ \eta_k \sqrt{\mathbb{E}\|\nabla_x \mathbb{E}_\Omega f(x_k, \omega) - d_k\|^2}D_{\mathcal{X}}$$

$$\leq (1 - \delta\eta_k)\big(\mathbb{E}F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big) + \frac{9D_{\mathcal{X}}\sqrt{Q}}{(\delta(k-1)+9)^{\frac{4}{3}}}$$

Now, note that:

$$\frac{\eta_k^2}{2} D_{\mathcal{X}}^2\big(L_f + \frac{\|A\|^2}{\beta_k}\big) = \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 L_f + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 \frac{\|A\|^2}{\beta_k}$$

$$= \frac{81/2}{(\delta(k-1)+9)^2} D_{\mathcal{X}}^2 L_f + \frac{81/2}{(\delta(k-1)+9)^{\frac{3}{2}}}\beta_0 D_{\mathcal{X}}^2\|A\|^2$$

$$\leq \frac{81/2}{(\delta(k-1)+9)^{\frac{4}{3}}} D_{\mathcal{X}}^2 L_f + \frac{81/2}{(\delta(k-1)+9)^{\frac{4}{3}}}\beta_0 D_{\mathcal{X}}^2\|A\|^2$$

Therefore:

$$\mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star \leq \left(1 - \frac{9\delta}{\delta(k-1)+9}\right)\big(F_{\beta_{k-1}}(x_k) - F^\star\big) + \frac{\frac{81}{2}D_{\mathcal{X}}^2(L_f + \beta_0\|A\|^2) + 9D_{\mathcal{X}}\sqrt{Q}}{(\delta(k-1)+9)^{\frac{4}{3}}}$$

For simplicity, let $C := \frac{81}{2}D_{\mathcal{X}}^2(L_f + \beta_0\|A\|^2) + 9D_{\mathcal{X}}\sqrt{Q}$ and $\mathcal{E}_{k+1} := \mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star$. Then, we need to solve the following recursive equation:

$$\mathcal{E}_{k+1} \leq \left(1 - \frac{9\delta}{\delta(k-1)+9}\right)\mathcal{E}_k + \frac{C}{(\delta(k-1)+9)^{\frac{4}{3}}} \tag{44}$$

Let the induction hypothesis for $k \geq 1$ be:

$$\mathcal{E}_{k+1} \leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1)+9)^{\frac{1}{3}}}$$

21

The base case $k = 1$ is trivial as from (44) we have $\mathcal{E}_2 \leq (1 - \delta)\mathcal{E}_1 + \frac{C}{9^{\frac{4}{3}}} \leq \mathcal{E}_1 + \frac{C}{9^{\frac{4}{3}}} \leq \mathcal{E}_1 + \frac{C}{\delta}$

For simplicity let $K := \delta(k - 1) + 9$. From Equation (34) we have $\mathcal{E}_2 \leq \frac{C}{(9)^{\frac{4}{3}}} < C$ as $9^{\frac{4}{3}} > 1$ Now:

$$\mathcal{E}_{k+1} \leq \left(1 - \frac{9\delta}{K}\right)\mathcal{E}_k + \frac{C}{(K)^{\frac{4}{3}}}$$

$$\leq \left(1 - \frac{9\delta}{K}\right)9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(K - \delta)^{\frac{1}{3}}} + \frac{C}{(K)^{\frac{4}{3}}}$$

$$\leq \left(1 - \frac{9\delta}{K}\right)9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(K - \delta)^{\frac{1}{3}}} + 9^{\frac{1}{3}}\delta\frac{\frac{C}{\delta} + \mathcal{E}_1}{K(K - \delta)^{\frac{1}{3}}}$$

$$\leq \left(1 - \frac{8\delta}{K}\right)9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(K - \delta)^{\frac{1}{3}}}$$

$$\leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(K + \delta)^{\frac{1}{3}}}$$

$\square$

Corollary' 5. *Assume that g is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 1 with approximate lmo (24), and modifying $\eta_k = \frac{9}{\delta(k-1)+9}$, $\beta_k = \frac{\beta_0}{\sqrt{\delta(k-1)+9}}$ and $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$ satisfies:*

$$\mathbb{E}F(x_{k+1}) - F^\star \leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k - 1) + 9)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{\delta(k-1)+9}},$$

We can optimize β_0 from this bound if δ is known.

Proof. If $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is L_g-Lipschitz continuous we get from (35):

$$g(Ax_{k+1}) \leq g_{\beta_k}(Ax_{k+1}) + \frac{\beta_k L_g^2}{2}$$

$$= g_{\beta_k}(Ax_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{\delta(k - 1) + 9}}.$$

We complete the proof by adding $\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - F^\star$ to both sides:

$$\mathbb{E}F(x_{k+1}, \omega) - F^\star \leq \mathbb{E}F_{\beta_k}(x_{k+1}) - F^\star + \frac{\beta_0 L_g^2}{2\sqrt{k+8}}$$

$$\leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k - 1) + 9)^{\frac{1}{3}}} + \frac{\beta_0 L_g^2}{2\sqrt{\delta(k - 1) + 9}}.$$

$\square$

Corollary' 6. *Assume that g is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 1 with approximate lmo (24), and modifying $\eta_k = \frac{9}{\delta(k-1)+9}$, $\beta_k = \frac{\beta_0}{\sqrt{\delta(k-1)+9}}$ and $\rho_k = \frac{4}{(\delta(k-2)+9)^{\frac{2}{3}}}$ satisfies:*

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \geq -\|y^\star\|\,\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K})$$

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \leq 9^{\frac{1}{3}}\frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k - 1) + 9)^{\frac{1}{3}}}$$

$$\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) \leq \frac{2\beta_0\|y^\star\|}{\sqrt{\delta(k - 1) + 9}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}}(\frac{C}{\delta} + \mathcal{E}_1)\beta_0}}{(\delta(k - 1) + 9)^{\frac{5}{12}}}$$

Proof. We adapt to our rate the proof technique of Theorem 4.3 in [41]. From the Lagrange saddle point theory, we know that the following bound holds $\forall x \in \mathcal{X}$ and $\forall r \in \mathcal{K}$:

$$f^\star \leq \mathcal{L}(x, r, y^\star) = \mathbb{E}_\Omega f(x, \omega) + \langle y_\star, Ax - r \rangle$$
$$\leq \mathbb{E}_\Omega f(x, \omega) + \|y_\star\| \|Ax - r\|,$$

Since $x_{k+1} \in \mathcal{X}$, we get after taking the expectation

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \geq -\mathbb{E} \min_{r \in \mathcal{K}} \|y^\star\| \|Ax_{k+1} - r\|$$
$$= -\|y^\star\| \mathbb{E} \mathrm{dist}(Ax_{k+1}, \mathcal{K}). \qquad (45)$$

This proves the first bound in Corollary 2.

The second bound directly follows by Theorem 9 as

$$\mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star \leq \mathbb{E}\mathbb{E}_\Omega f(x_{k+1}, \omega) - f^\star + \frac{1}{2\beta_k} \mathbb{E} \left[\mathrm{dist}(Ax_{k+1}, \mathcal{K})\right]^2$$
$$\leq \mathbb{E} F_{\beta_k}(x_{k+1}) - F^\star$$
$$\leq 9^{\frac{1}{3}} \frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1) + 9)^{\frac{1}{3}}}.$$

Now, we combine this with (45), and we get

$$-\|y^\star\| \mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) + \frac{1}{2\beta_k} \mathbb{E}\left[\mathrm{dist}(Ax_{k+1}, \mathcal{K})\right]^2 \leq 9^{\frac{1}{3}} \frac{\frac{C}{\delta} + \mathcal{E}_1}{(\delta(k-1) + 9)^{\frac{1}{3}}}$$

This is a second order inequality in terms of $\mathbb{E}\mathrm{dist}(Ax_k, \mathcal{K})$. Solving this inequality, we get

$$\mathbb{E}\mathrm{dist}(Ax_{k+1}, \mathcal{K}) \leq \frac{2\beta_0 \|y^\star\|}{\sqrt{\delta(k-1) + 9}} + \frac{2\sqrt{2 \cdot 9^{\frac{1}{3}} (\frac{C}{\delta} + \mathcal{E}_1)\beta_0}}{(\delta(k-1) + 9)^{\frac{5}{12}}}.$$

$\square$