Conditional gradient-based method for bilevel optimization with convex lower-level problem

Ruichen Jiang RJIANG@UTEXAS.EDU
The University of Texas at Austin

Nazanin Abolfazli NAZANINABOLFAZLI@EMAIL.ARIZONA.EDU
The University of Arizona

Aryan Mokhtari MOKHTARI@AUSTIN.UTEXAS.EDU
The University of Texas at Austin

Erfan Yazdandoost Hamedani ERFANY@ARIZONA.EDU
The University of Arizona

Abstract

In this paper, we study simple bilevel optimization problems, where we minimize a smooth objective function over the optimal solution set of another convex constrained optimization problem. Several iterative methods have been developed for tackling this class of problems. Alas, their convergence guarantees are not satisfactory as they are either asymptotic for the upper-level objective, or the convergence rates are slow and sub-optimal. To address this issue, in this paper, we introduce a conditional gradient-based (CG-based) method to solve the considered problem. The main idea is to locally approximate the solution set of the lower-level problem via a cutting plane, and then run a CG-type update to decrease the upper-level objective. When the upper-level objective is convex, we show that our method requires $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations to find a solution that is ϵ_f-optimal for the upper-level objective and ϵ_g-optimal for the lower-level objective. Moreover, when the upper-level objective is non-convex, our method requires $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$ iterations to find an (ϵ_f, ϵ_g)-stationary solution. To the best of our knowledge, our method achieves the best-known iteration complexity for the considered bilevel problem.

1. Introduction

Bilevel optimization is a form of optimization where one problem is embedded within another. It captures a hierarchical structure, where an *upper-level* function is minimized over the solution set of a *lower-level* problem. This class of problems has attracted great attention due to their applications in hyper-parameter optimization [15, 46], meta-learning [4, 42], and reinforcement learning [23, 28], to name a few. In this paper, we focus on a specific form of bilevel optimization formally defined as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \; g(\mathbf{z}), \tag{1}$$

where $\mathcal{Z}$ is a compact convex set and $f, g : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable functions on an open set containing $\mathcal{Z}$. We assume that g is convex but not necessarily strictly convex, and hence the lower-level problem in (1) could have multiple optimal solutions. We remark that Problem (1) is often referred to as the "simple bilevel problem" in the literature [12, 14, 47] to differentiate it from the more general settings where the lower-level problem is parameterized by some upper-level variables.

© R. Jiang, N. Abolfazli, A. Mokhtari & E. Yazdandoost Hamedani.

This class of bilevel problems appears in several settings such as hyperparameter selection [16, 53] and fair classification [17, 55]; we review a few of them in Appendix A.

The key challenge to solve Problem (1) stems from the fact that its feasible set—the solution set of the lower-level problem—does not admit a simple characterization and is not explicitly given. This rules out the possibility of applying projection-based methods as well as the conditional gradient (CG) method, since projection onto or minimizing a linear objective over the feasible set is computationally intractable. An alternative scheme is reformulating Problem (1) as a constrained optimization problem with the functional constraint $g(\mathbf{x}) \leq g^*$ and applying primal-dual methods, where g^* is the optimal value of the lower-level problem. However, a critical issue is that the resulting problem does not satisfy strict feasibility and hence the Slater's condition fails, which is required for most primal-dual methods. Even relaxing the constraint ($g(\mathbf{x}) \leq g^* + \epsilon$) to ensure strict feasibility would inevitably lead to numerical issues (see more discussions in Appendix F).

Therefore, Problem (1) cannot be simply treated as a classic constrained optimization problem and calls for new theories and algorithms tailored to its hierarchical structure [9, 48, 50, 52]. More recently, there has been a surge of interest in establishing non-asymptotic convergence rates for Problem (1). One of the first methods of this kind is the minimal norm gradient (MNG) method by Beck and Sabach [3]. When f is strongly-convex and g is convex and smooth, they showed that MNG converges asymptotically to the optimal solution and achieves a complexity bound of $\mathcal{O}(1/\epsilon^2)$ in terms of the lower-level objective. Subsequently, the BiG-SAM method was proposed by Sabach and Shtern [45] and it was shown to achieve a complexity of $\mathcal{O}(1/\epsilon)$ for the lower-level problem; see also Shehu et al. [47] for a related method. Malitsky [38] studied a version of Tseng's accelerated gradient method that obtains a convergence rate of $o(1/k)$ for the lower-level problem. When f and g are convex and Lipschitz, Kaushik and Yousefian [26] studied iterative regularization and showed a convergence rate of $\mathcal{O}(1/k^{0.5-b})$ for the upper-level objective and a rate of $\mathcal{O}(1/k^b)$ for the lower-level, where $b \in (0, 0.5)$ is a user-defined parameter. Several works also extended this method to stochastic [1, 2] and distributed [27, 54] settings.

Contributions. As discussed, prior works only establish convergence rates for the lower-level problem, while the rate for the upper-level is missing. The only exception is the work by Kaushik and Yousefian [26], but they consider a different setting where both upper-level and lower-level functions are Lipschitz and possibly non-smooth, which results in slow convergence rates; see Table 1 in the Appendix. Our main contribution is presenting the conditional gradient-based bilevel (CG-BiO) method with tight non-asymptotic guarantees for both upper- and lower-level objectives. At each iteration, CG-BiO uses a cutting plane to locally approximate the solution set of the lower-level problem, and applies a CG-type update on the upper-level objective. Our theoretical guarantees for CG-BiO are:

- When the upper-level function f is convex, we show that CG-BiO can find $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) - f^* \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations, where f^* is the optimal value of Problem (1) and g^* is the optimal value of the lower-level problem. This guarantee matches the best-known results in terms of the lower-level objective and are optimal for bilevel projection-free methods.

- When f is non-convex, CG-BiO finds $\hat{\mathbf{x}}$ that satisfies $\mathcal{G}(\hat{\mathbf{x}}) \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f \epsilon_g)\})$ iterations, where $\mathcal{G}(\hat{\mathbf{x}})$ is the Frank-Wolfe (FW) gap function (cf. (3)).

Additional related work. In the general form of bilevel problems, the upper-level function f may also depend on an additional variable $\mathbf{w} \in \mathbb{R}^m$ that in turn influences the lower-level problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{w}\in\mathbb{R}^m} f(\mathbf{x}, \mathbf{w}) \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z}, \mathbf{w}). \tag{2}$$

Problem (2) has been studied deeply and we refer readers to the extensive survey by [11]. We can also see its close connection with the simple bilevel problem we study in this paper, as Problem (2) reduces to Problem (1) for any fixed $\mathbf{w}$. In recent years, gradient-based methods for Problem (2) have become increasingly popular including implicit differentiation [13, 18, 25, 41] and iterative differentiation [37]. However, most existing methods work under the assumption that the lower-level problem is strongly convex in $\mathbf{z}$ for any $\mathbf{w}$ and thus has a unique minimum. More relevant to our work, some concurrent papers consider the case where the lower-level problem can have multiple minima [31–34, 49]. As they consider a more general problem than ours, their theoretical results are also weaker, providing only asymptotic convergence guarantees or slower rates. In this paper, we develop a new approach for solving the bilevel optimization problem in (1) using a fundamentally different perspective.

2. Assumptions and definitions

In this section, we state the required assumptions and notions of optimality that we use for our theoretical results.

We focus on the case where the lower-level function g is smooth and convex, while the upper-level function f is smooth but not necessarily convex. Formally, we make the following assumptions.

Assumption 1 *Let $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^n$ and $\|\cdot\|_*$ be its dual norm. We assume*

(i) $\mathcal{Z} \subset \mathbb{R}^n$ is convex and compact with diameter D, i.e., $\|\mathbf{x}-\mathbf{y}\| \le D$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{Z}$.

(ii) g is convex and continuously differentiable on an open set containing $\mathcal{Z}$, and its gradient is Lipschitz with constant L_g, i.e.,$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|_* \le L_g\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{Z}$.

(iii) f is continuously differentiable and its gradient is Lipschitz with constant L_f.

Throughout the paper, we use $g^* \triangleq \min_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$ and $\mathcal{X}_g^* \triangleq \operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$ to denote the optimal value and the optimal solution set of the lower-level problem, respectively. Note that by Assumption 1, the set $\mathcal{X}_g^*$ is nonempty, compact and convex, but in general not a singleton as g could have multiple minima on $\mathcal{Z}$. Moreover, we use f^* to denote the optimal value and $\mathbf{x}^*$ to denote an optimal solution of Problem (1), which are guaranteed to exist as f is continuous and $\mathcal{X}_g^*$ is compact.

For generality, we allow different target accuracies ϵ_f and ϵ_g for the upper-level and lower-level problems, respectively, and define an (ϵ_f, ϵ_g)-solution as follows.

Definition 1 ((ϵ_f, ϵ_g)-solution) *When f is convex, a point $\hat{\mathbf{x}} \in \mathcal{Z}$ is (ϵ_f, ϵ_g)-optimal for the bilevel problem in (1) if $f(\hat{\mathbf{x}}) - f^* \le \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \le \epsilon_g$. When f is non-convex, $\hat{\mathbf{x}} \in \mathcal{Z}$ is (ϵ_f, ϵ_g)-stationary if $\mathcal{G}(\hat{\mathbf{x}}) \le \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \le \epsilon_g$, where $\mathcal{G}(\hat{\mathbf{x}})$ is the FW gap defined by*

$$\mathcal{G}(\hat{\mathbf{x}}) \triangleq \max_{\mathbf{s}\in\mathcal{X}_g^*} \{\langle \nabla f(\hat{\mathbf{x}}), \hat{\mathbf{x}} - \mathbf{s}\rangle\}. \tag{3}$$

Remark 2 *The FW gap is a standard performance metric for conditional gradient methods [24, 29]. For any feasible point $\hat{\mathbf{x}} \in \mathcal{X}_g^*$, it is known that $\hat{\mathbf{x}}$ is a first-order stationary point if and only if $\mathcal{G}(\hat{\mathbf{x}}) = 0$.*

Algorithm 1 Conditional gradient-based bilevel (CG-BiO)

Input: Target accuracies $\epsilon_f, \epsilon_g > 0$, stepsizes $\{\gamma_k\}_k$
Initialization: Initialize $\mathbf{x}_0 \in \mathcal{Z}$ such that $0 \leq g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$
for $k = 0, \ldots, K$ **do**
  Compute $\mathbf{s}_k \leftarrow \operatorname{argmin}_{\mathbf{s}\in\mathcal{X}_k} \langle \nabla f(\mathbf{x}_k), \mathbf{s}\rangle$
  where $\mathcal{X}_k \triangleq \{\mathbf{s} \in \mathcal{Z} : \langle \nabla g(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k\rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)\}$
  if $\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s}_k\rangle \leq \epsilon_f$ and $\langle \nabla g(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s}_k\rangle \leq \epsilon_g/2$ **then**
    Return $\mathbf{x}_k$ and STOP
  else
    $\mathbf{x}_{k+1} \leftarrow (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k$
  end if
end for

3. Conditional gradient-based method for bilevel optimizaiton

Before stating our proposed method, we start by the standard CG method for solving Problem (1). Recall that $\mathcal{X}_g^*$ denotes the solution set of the lower-level problem. If we assume $\mathbf{x}_0 \in \mathcal{X}_g^*$, then the update of CG at iteration k is given by

$$\mathbf{x}_{k+1} = (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k \quad \text{where } \mathbf{s}_k = \operatorname*{argmin}_{\mathbf{s}\in\mathcal{X}_g^*} \langle \nabla f(\mathbf{x}_k), \mathbf{s}\rangle\,, \tag{4}$$

and $\gamma_k \in [0, 1]$ is the stepsize. However, as we discussed earlier, the main challenge here is that the solution set $\mathcal{X}_g^*$ for the lower-level problem is not explicitly given, and hence the linear minimization required in (4) is computationally intractable. Moreover, the standard CG method needs to be initialized with a feasible point. In this case, $\mathbf{x}_0$ has to be an optimal solution of the lower-level problem, which is hard to guarantee in general—in finite number of iterations one may not be able to find an *exact* optimal solution for the lower-level problem. Similar issues also hold if we try to use projection-based methods such as projected gradient descent to solve Problem (1).

Our key idea is to run the CG update over a local approximation set $\mathcal{X}_k$ at the k-th iteration in place of the more complicated set $\mathcal{X}_g^*$. To this end, we borrow the idea of *cutting plane* from the optimization literature [6] and let $\mathcal{X}_k$ be the intersection of $\mathcal{Z}$ and the halfspace $\mathcal{H}_k$:

$$\mathcal{X}_k \triangleq \mathcal{Z} \cap \mathcal{H}_k, \text{where } \mathcal{H}_k = \{\mathbf{s} \in \mathbb{R}^n : \langle \nabla g(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k\rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)\}. \tag{5}$$

We can see that $\mathcal{X}_k$ is potentially more tractable than $\mathcal{X}_g^*$, as the difficult nonlinear inequality $g(\mathbf{x}) \leq g^*$ is replaced by a single linear inequality. Also, by using the convexity of g, we can show that the hyperplane $\mathcal{H}_k$ eliminates those points that are known to have a larger value of than $g(\mathbf{x}_0)$. Thus, if we initialize our algorithm such that $\mathbf{x}_0$ is near-optimal, the linear inequality in (5) ensures improvement in terms of the lower-level function. Further, this also implies that $\mathcal{X}_k$ contains the solution set $\mathcal{X}_g^*$, so we are guaranteed to make progress on the upper-level objective f. We justify this observation in the following lemma.

Lemma 3 *Recall $\mathcal{X}_g^*$ as the solution set for the lower-level problem in (1) and recall the definition of the set $\mathcal{X}_k$ in (5). Then, for any $k \geq 0$, we have $\mathcal{X}_g^* \subseteq \mathcal{X}_k$.*

Now we are ready to state our CG-BiO method. We first initialize $\mathbf{x}_0 \in \mathcal{Z}$ as a near-optimal solution for the lower-level problem, i.e., $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$ for some prescribed accuracy ϵ_g. This can be achieved by running the standard CG method on the lower-level problem, which requires at most $\mathcal{O}(1/\epsilon_g)$ iterations. Once the initialization step is done, we simply run CG with respect to the approximation sets $\mathcal{X}_k$. More precisely, at the k-th iteration, we solve the following subproblem over the set $\mathcal{X}_k$ defined in (5):

$$\mathbf{s}_k = \operatorname*{argmin}_{\mathbf{s} \in \mathcal{X}_k} \ \langle \nabla f(\mathbf{x}_k), \mathbf{s} \rangle \,, \tag{6}$$

and update the iterate by $\mathbf{x}_{k+1} = (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k$ with stepsize $\gamma_k \in [0,1]$. Here, we assume access to a linear optimization oracle that returns the solution of the subproblem in (6), which is standard for projection-free methods [24, 29, 39]. In particular, if $\mathcal{Z}$ can be described by a system of linear inequalities, then the subproblem in (6) corresponds to a linear program and can be solved efficiently by a standard solver. We repeat the process above until we reach an accuracy of ϵ_f for the upper-level objective and an accuracy of ϵ_g for the lower-level objective. The steps of our proposed CG-BiO method are summarized in Algorithm 1.

4. Convergence analysis

In this section, we analyze the iteration complexity of our CG-BiO method. We first consider the case where the upper-level function f is convex. In this case, we choose the stepsize as $\gamma_k = 2/(k+2)$, which is typical in the standard CG method [24].

Theorem 4 (Convex upper-level) *Suppose that Assumption 1 holds and f is convex. Let $\{\mathbf{x}_k\}_{k=0}^K$ be the sequence generated by Algorithm 1 with stepsize $\gamma_k = 2/(k+2)$ for $k \geq 0$. Then we have*

$$f(\mathbf{x}_K) - f^* \leq \frac{2L_f D^2}{K+1} \quad \textit{and} \quad g(\mathbf{x}_K) - g^* \leq \frac{2L_g D^2}{K+1} + \frac{1}{2}\epsilon_g.$$

Theorem 4 shows that the gap of the upper-level objective can be upper bounded by $\mathcal{O}(1/K)$, similar to the convergence bound of standard CG. At the same time, the gap of the lower-level objective can also be controlled by a term of order $\mathcal{O}(1/K)$ in addition to the initial error $\epsilon_g/2$. As a corollary, Algorithm 1 will return an (ϵ_f, ϵ_g)-optimal solution when the number of iterations K exceeds

$$\max\left\{\frac{2L_f D^2}{\epsilon_f} - 1, \frac{4L_g D^2}{\epsilon_g} - 1\right\} = \mathcal{O}\left(\max\left\{\frac{1}{\epsilon_f}, \frac{1}{\epsilon_g}\right\}\right).$$

Our complexity bound improves over the result by Kaushik and Yousefian [26], who consider a different setup where both the upper-level and lower-level functions are Lipschitz but not necessarily smooth. Also, comparing with existing works in the same setup, our convergence rate for the lower-level objective matches those in [38, 45], while we also provide a non-asymptotic convergence bound for the upper-level objective. To the best of our knowledge, our result provides the best-known bound for the considered setting. We also remark that our rate is tight at least within the family of projection-free methods, since it is known that their worst-case complexity is $\Theta(1/\epsilon_f)$ even for a single-level problem [24, 30].

Remark 5 *As the initialization step requires $\mathcal{O}(1/\epsilon_g)$ iterations, this additional cost will not be the dominant term in the final complexity. The same applies for the non-convex setting below.*

Now we turn to the case where f is non-convex. In this case, we choose the stepsize as a constant depending on the target accuracies as well as the problem parameters.

Theorem 6 (Non-convex upper-level) *Suppose that Assumption 1 holds. Let* $\{\mathbf{x}_k\}_{k=0}^{K-1}$ *be the sequence generated by Algorithm 1 with stepsize* $\gamma_k = \min\left\{\frac{\epsilon_f}{L_f D^2}, \frac{\epsilon_g}{L_g D^2}\right\}$ *for all* $k \geq 0$. *Define* $\underline{f} = \min_{\mathbf{x}\in Z} f(\mathbf{x})$. *Then for* $K \geq \max\left\{\frac{2L_f D^2(f(\mathbf{x}_0)-\underline{f})}{\epsilon_f^2}, \frac{2L_g D^2(f(\mathbf{x}_0)-\underline{f})}{\epsilon_f \epsilon_g}\right\}$, *there exists* $k^* \in \{0, 1, \ldots, K-1\}$ *such that* $\mathcal{G}(\mathbf{x}_{k^*}) \leq \epsilon_f$ *and* $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$.

As a corollary of Theorem 6, the number of iterations required to find an (ϵ_f, ϵ_g)-optimal solution can be upper bounded by $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$. We note that the dependence on the upper-level accuracy ϵ_f also matches that in the standard CG method for a single-level problem [29, 39].

5. Conclusion

In this paper, we proposed a conditional gradient-based method to solve a class of bilevel optimization problems. We closed an important gap in the existing literature by providing a tight non-asymptotic complexity bound for the upper-level objective. Specifically, we proved that our CG-BiO algorithm can find an (ϵ_f, ϵ_g)-optimal solution after at most $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations when the upper-level objective f is convex, and after at most $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$ iterations when f is non-convex. To the best of our knowledge, our work presents the best iteration complexity in the considered bilevel problem.

References

[1] Mostafa Amini and Farzad Yousefian. An iterative regularized mirror descent method for ill-posed nondifferentiable stochastic optimization. *arXiv preprint arXiv:1901.09506*, 2019.

[2] Mostafa Amini and Farzad Yousefian. An iterative regularized incremental projected subgradient method for a class of bilevel optimization problems. In *2019 American Control Conference (ACC)*, pages 4069–4074. IEEE, 2019.

[3] Amir Beck and Shoham Sabach. A first order method for finding minimal norm-like solutions of convex optimization problems. *Mathematical Programming*, 147(1-2):25–46, 2014.

[4] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

[5] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017. ISSN 1436-4646. doi: 10.1007/s10107-016-1091-6. URL https://doi.org/10.1007/s10107-016-1091-6.

[6] Stephen Boyd and Lieven Vandenberghe. Localization and cutting-plane methods. https://web.stanford.edu/class/ee364b/lectures/localization_methods_notes.pdf, 2018. URL https://web.stanford.edu/class/ee364b/lectures/localization_methods_notes.pdf.

[7] J. V. Burke and M. C. Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993. doi: 10.1137/0331063. URL https://doi.org/10.1137/0331063.

[8] James V. Burke and Sien Deng. Weak sharp minima revisited, part ii: application to linear regularity and error bounds. *Mathematical Programming*, 104(2):235–261, 2005. ISSN 1436-4646. doi: 10.1007/s10107-005-0615-2. URL https://doi.org/10.1007/s10107-005-0615-2.

[9] Alexandre Cabot. Proximal point algorithm controlled by a slowly vanishing term: Applications to hierarchical minimization. *SIAM Journal on Optimization*, 15(2):555–572, 2005. doi: 10.1137/S105262340343467X. URL https://doi.org/10.1137/S105262340343467X.

[10] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1):253–287, 2016.

[11] Stephan Dempe. *Bilevel Optimization: Theory, Algorithms, Applications and a Bibliography*, pages 581–672. Springer International Publishing, Cham, 2020. ISBN 978-3-030-52119-6. doi: 10.1007/978-3-030-52119-6_20.

[12] Stephen Dempe, Nguyen Dinh, and Joydeep Dutta. *Optimality Conditions for a Simple Convex Bilevel Programming Problem*, pages 149–161. Springer New York, New York, NY, 2010. ISBN 978-1-4419-0437-9. doi: 10.1007/978-1-4419-0437-9_7. URL https://doi.org/10.1007/978-1-4419-0437-9_7.

[13] Justin Domke. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 318–326, 2012. URL https://proceedings.mlr.press/v22/domke12.html.

[14] Joydeep Dutta and Tanushree Pandit. *Algorithms for Simple Bilevel Programming*, pages 253–291. Springer International Publishing, Cham, 2020. ISBN 978-3-030-52119-6. doi: 10.1007/978-3-030-52119-6_9. URL https://doi.org/10.1007/978-3-030-52119-6_9.

[15] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1568–1577, 2018. URL https://proceedings.mlr.press/v80/franceschi18a.html.

[16] Lucy L Gao, Jane Ye, Haian Yin, Shangzhi Zeng, and Jin Zhang. Value function based difference-of-convex algorithm for bilevel hyperparameter selection problems. In *Proceedings of the 39th International Conference on Machine Learning*, pages 7164–7182, 2022. URL https://proceedings.mlr.press/v162/gao22j.html.

[17] Chengyue Gong, Xingchao Liu, and Qiang Liu. Bi-objective trade-off with dynamic barrier gradient descent. In *Advances in Neural Information Processing Systems*, 2021.

[18] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[19] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[20] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

[21] Erfan Yazdandoost Hamedani and Necdet Serhat Aybat. A primal-dual algorithm with line search for general convex-concave saddle point problems. *SIAM Journal on Optimization*, 31 (2):1299–1329, 2021.

[22] Niao He, Anatoli Juditsky, and Arkadi Nemirovski. Mirror prox algorithm for multi-term composite minimization and semi-separable problems. *Computational Optimization and Applications*, 61(2):275–319, 2015.

[23] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

[24] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013. URL https://proceedings.mlr.press/v28/jaggi13.html.

[25] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892, 2021.

[26] Harshal D. Kaushik and Farzad Yousefian. A method with convergence rates for optimization problems with variational inequality constraints. *SIAM Journal on Optimization*, 31 (3):2171–2198, 2021. doi: 10.1137/20M1357378. URL https://doi.org/10.1137/20M1357378.

[27] Harshal D. Kaushik and Farzad Yousefian. Distributed optimization for problems with variational inequality constraints. *arXiv preprint arXiv:2105.14205*, 2021.

[28] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

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

[30] Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

[31] Junyi Li, Bin Gu, and Heng Huang. Improved bilevel model: Fast and optimal algorithm with theoretical guarantee. *arXiv preprint arXiv:2009.00690*, 2020.

[32] Risheng Liu, Pan Mu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://proceedings.mlr.press/v119/liu20l.html.

[33] Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/liu21o.html.

[34] Risheng Liu, Yaohua Liu, Shangzhi Zeng, and Jin Zhang. Towards gradient-based bilevel optimization with non-convex followers and beyond. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/48bea99c85bcbaaba618ba10a6f69e44-Paper.pdf.

[35] Stanisław Łojasiewicz. Sur la problème de la division. *Studia Mathematica*, 18:87–136, 1959.

[36] Zhi-Quan Luo and Jong-Shi Pang. Error bounds for analytic systems and their applications. *Mathematical Programming*, 67(1):1–28, 1994. ISSN 1436-4646. doi: 10.1007/BF01582210. URL https://doi.org/10.1007/BF01582210.

[37] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2113–2122, 2015. URL https://proceedings.mlr.press/v37/maclaurin15.html.

[38] Yura Malitsky. Chambolle-pock and tseng's methods: relationship and extension to the bilevel optimization. *arXiv preprint arXiv:1706.02602*, 2017.

[39] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[40] Jong-Shi Pang. Error bounds in mathematical programming. *Mathematical Programming*, 79(1):299–332, 1997. ISSN 1436-4646. doi: 10.1007/BF02614322. URL https://doi.org/10.1007/BF02614322.

[41] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, 2016. URL http://proceedings.mlr.press/v48/pedregosa16.html.

[42] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

[43] Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020. doi: 10.1137/18M1224568.

[44] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models, 2021.

[45] Shoham Sabach and Shimrit Shtern. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.

[46] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732, 2019.

[47] Yekini Shehu, Phan Tu Vuong, and Alain Zemkoho. An inertial extrapolation method for convex simple bilevel optimization. *Optimization Methods and Software*, 36(1):1–19, 2021.

[48] Mikhail Solodov. An explicit descent method for bilevel convex optimization. *Journal of Convex Analysis*, 14(2):227, 2007.

[49] Daouda Sow, Kaiyi Ji, Ziwei Guan, and Yingbin Liang. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

[50] Hong-Kun Xu. Viscosity approximation methods for nonexpansive mappings. *Journal of Mathematical Analysis and Applications*, 298(1):279–291, 2004.

[51] Yangyang Xu. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. *Mathematical Programming*, 185(1):199–244, 2021.

[52] Isao Yamada. The hybrid steepest-descent method for variational inequalityproblems over the intersection of the fixed-point sets of nonexpansive mappings. In *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 473–504. North-Holland, 2001.

[53] Jane J. Ye, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *Mathematical Programming*, 2022. URL https://doi.org/10.1007/s10107-022-01888-3.

[54] Farzad Yousefian. Bilevel distributed optimization in directed networks. In *2021 American Control Conference (ACC)*, pages 2230–2235, 2021. doi: 10.23919/ACC50511.2021.9483429.

[55] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019. URL http://jmlr.org/papers/v20/18-262.html.

Appendix

Appendix A. Motivating examples

Many practical machine learning applications consist of a primal objective g, such as the training loss, and a secondary objective f, such as a regularization term or auxiliary loss. In this case, a natural approach is to fully optimize the primal objective and use the secondary objective as a criterion to select one of the optimal solutions. Such kind of problem, also known as lexicographic optimization [17], can be exactly formulated as the simple bilevel problem in (1).

To be concrete, consider an empirical risk minimization problem in the form of $\min_{\boldsymbol{\beta}\in\mathcal{Z}} \ell_{\mathrm{tr}}(\boldsymbol{\beta}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}_i;\boldsymbol{\beta}), y_i)$, where $\mathcal{Z} \subseteq \mathbb{R}^d$ is the constraint set, h is the learning model parametrized by $\boldsymbol{\beta}$, and ℓ is the loss function corresponding to input $\mathbf{x}_i$ and its associated label y_i. Typically, the primal objective g is chosen as the training loss $\ell_{\mathrm{tr}}(\boldsymbol{\beta})$ and different choices of the secondary objective give rise to different problems. We provide several examples in the following.

Example 1 (Ill-posed optimization) *Without an explicit regularization, the empirical risk minimization problem above can be ill-posed, i.e., it has multiple optimal solutions or is sensitive to small perturbation in the input data. To tackle this issue, we can introduce a regularization term $\mathcal{R}(\cdot)$ as the secondary objective, leading to the following bilevel problem:*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \ f(\boldsymbol{\beta}) \triangleq \mathcal{R}(\boldsymbol{\beta}) \quad \textit{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \ g(\mathbf{z}) \triangleq \ell_{\mathrm{tr}}(\mathbf{z}).$$

In particular, by choosing $\mathcal{R}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2$ we can find the minimal ℓ_2-norm solution.

Example 2 (Hyperparameter selection) *Most machine learning algorithms require careful hyperparameter tuning, and a common strategy is to select the set of hyperparameters that also minimizes the loss over some validation set $\mathcal{D}_{\mathrm{val}}$. In this case, the validation loss $\ell_{\mathrm{val}}(\boldsymbol{\beta})$ serves as the secondary objective, leading to the following bilevel problem:*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \ \ell_{\mathrm{val}}(\boldsymbol{\beta}) \quad \textit{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \ \ell_{\mathrm{tr}}(\mathbf{z};\boldsymbol{\lambda}), \tag{7}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ denotes the hyperparameters. Note that for any fixed $\boldsymbol{\lambda}$, Problem (7) becomes an instance of the simple bilevel problem in (1). In particular, we can then use grid search or random search over $\boldsymbol{\lambda}$ when the number of hyperparameters is small.

Example 3 (Fair classification) *Standard training procedures could lead to a model that discriminates against certain society groups. To alleviate this issue, we can use a fairness metric as a secondary objective to promote fairness in the decision of the model. For instance, we may consider the following bilevel problem:*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \ (\mathrm{cov}(h(\mathbf{x};\boldsymbol{\beta}), \mathbf{v}))^2 \quad \textit{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \ \ell_{\mathrm{tr}}(\mathbf{z}),$$

where we use the covariance between the output of the model $h(\mathbf{x};\boldsymbol{\beta})$ and the sensitive features $\mathbf{v}$ as the fairness metric [17, 55].

Appendix B. Summary table

Table 1: Summary of bilevel optimization algorithms. The abbreviations "SC", "C", and "non-C" stand for "strongly convex", "convex", and "non-convex", respectively.

References	Upper level	Lower level		Convergence		Projection free?
	Objective f	Objective g	Feasible set $\mathcal{Z}$	Upper level	Lower level	
MNG [3]	SC, differentiable	C, smooth	Closed	Asymptotic	$\mathcal{O}(1/\epsilon^2)$	✗
BiG-SAM [45]	SC, smooth	C, composite	Closed	Asymptotic	$\mathcal{O}(1/\epsilon)$	✗
Tseng's method [38]	C, composite	C, composite	Closed	Asymptotic	$o(1/\epsilon)$	✗
a-IRG [26]	C, Lipschitz	C, Lipschitz	Closed	$\mathcal{O}(\max\{1/\epsilon_f^4, 1/\epsilon_g^4\})$		✗
Ours	C, smooth	C, smooth	Compact	$\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$		✓
Ours	Non-C, smooth	C, smooth	Compact	$\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$		✓

Appendix C. Supporting lemmas

C.1. Proof of Lemma 3

Let $\mathbf{x}_g^*$ be any point in $\mathcal{X}_g^*$, i.e., any optimal solution of the lower-level problem. By definition, we have $g(\mathbf{x}_g^*) = g^*$. Since g is convex and $g^* \leq g(\mathbf{x}_0)$, we have

$$g(\mathbf{x}_0) - g(\mathbf{x}_k) \geq g^* - g(\mathbf{x}_k) = g(\mathbf{x}_g^*) - g(\mathbf{x}_k) \geq \left\langle \nabla g(\mathbf{x}_k), \mathbf{x}_g^* - \mathbf{x}_k \right\rangle,$$

which implies $\mathbf{x}_g^* \in \mathcal{X}_k$. Hence, we conclude that $\mathcal{X}_g^* \subseteq \mathcal{X}_k$.

C.2. Improvement in one step

The following lemma characterizes the improvement of both the upper-level and lower-level objective values after one step of Algorithm 1.

Lemma 7 *Let $\{\mathbf{x}_k\}_{k=0}^K$ be the sequence generated by Algorithm 1. Suppose Assumption 1 holds, then for any $k \geq 0$ we have*

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \gamma_k \mathcal{G}(\mathbf{x}_k) + \frac{1}{2}\gamma_k^2 L_f D^2, \tag{8}$$

$$g(\mathbf{x}_{k+1}) \leq (1-\gamma_k) g(\mathbf{x}_k) + \gamma_k g(\mathbf{x}_0) + \frac{1}{2}\gamma_k^2 L_g D^2, \tag{9}$$

Proof

Since the gradient of f is L_f-Lipschitz and $\mathcal{Z}$ is bounded with diameter D, we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{2} L_f \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &= f(\mathbf{x}_k) + \gamma_k \langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_f \gamma_k^2 \|\mathbf{s}_k - \mathbf{x}_k\|^2 \\ &\leq f(\mathbf{x}_k) + \gamma_k \langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_f \gamma_k^2 D^2. \end{aligned} \tag{10}$$

Now using the definition of $\mathbf{s}_k$ in (6), the definition of $\mathcal{G}(\mathbf{x})$ in (3) and Lemma 3, we obtain

$$\langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle = \min_{\mathbf{s} \in \mathcal{X}_k} \langle \nabla f(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k \rangle \leq \min_{\mathbf{s} \in \mathcal{X}_g^*} \langle \nabla f(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k \rangle = -\mathcal{G}(\mathbf{x}_k). \tag{11}$$

Then (8) follows from (10) and (11).

Similarly, since the gradient of g is L_g-Lipschitz, we have

$$g(\mathbf{x}_{k+1}) \leq g(\mathbf{x}_k) + \gamma_k \langle \nabla g(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_g \gamma_k^2 D^2. \tag{12}$$

Moreover, since $\mathbf{s}_k \in \mathcal{X}_k$, from the definition of $\mathcal{X}_k$ in (6) we get $\langle \nabla g(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)$. Combining this with (12) leads to (9). ■

Appendix D. Proof of the main theorems

D.1. Proof of Theorem 4

We first prove the convergence rate of the upper-level objective f, which largely mirrors the standard analysis of the CG method [24]. Since $\mathbf{x}^* \in \mathcal{X}_g^*$ and f is convex, from the definition of $\mathcal{G}(\mathbf{x}_k)$ in (3) we have

$$\mathcal{G}(\mathbf{x}_k) = \max_{\mathbf{s} \in \mathcal{X}_g^*} \{ \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s} \rangle \} \geq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle \geq f(\mathbf{x}_k) - f^*. \tag{13}$$

Subtracting f^* from both sides of (8) in Lemma 7 and using (13), we obtain that

$$f(\mathbf{x}_{k+1}) - f^* \leq (1 - \gamma_k)(f(\mathbf{x}_k) - f^*) + \frac{1}{2} \gamma_k^2 L_f D^2. \tag{14}$$

Now define $A_k = k(k+1)$. By substituting $\gamma_k = 2/(k+2)$ and multiplying both sides of (14) by A_{k+1}, we get

$$A_{k+1}(f(\mathbf{x}_{k+1}) - f^*) \leq A_k(f(\mathbf{x}_k) - f^*) + \frac{2(k+1)}{k+2} L_f D^2 \leq A_k(f(\mathbf{x}_k) - f^*) + 2L_f D^2.$$

Hence, if follows from induction that

$$A_K(f(\mathbf{x}_K) - f^*) \leq A_0(f(\mathbf{x}_0) - f^*) + 2KL_f D^2 \quad \Rightarrow \quad f(\mathbf{x}_K) - f^* \leq \frac{2KL_f D^2}{A_k} = \frac{2L_f D^2}{K+1}.$$

This completes the first part of the proof.

The proof for the lower-level problem follows from similar arguments. By subtracting $g(\mathbf{x}_0)$ from both sides of (9) in Lemma 7, we have

$$g(\mathbf{x}_{k+1}) - g(\mathbf{x}_0) \leq (1 - \gamma_k)(g(\mathbf{x}_k) - g(\mathbf{x}_0)) + \frac{1}{2} \gamma_k^2 L_g D^2. \tag{15}$$

By substituting $\gamma_k = 2/(k+2)$ and multiplying both sides of (15) by A_{k+1}, we obtain

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}_0)) \leq A_k(g(\mathbf{x}_k) - g(\mathbf{x}_0)) + 2L_g D^2.$$

Hence, if follows from induction that

$$A_K(g(\mathbf{x}_K) - g(\mathbf{x}_0)) \leq 2KL_gD^2 \quad \Rightarrow \quad g(\mathbf{x}_K) - g(\mathbf{x}_0) \leq \frac{2KL_gD^2}{A_k} = \frac{2L_gD^2}{K+1}.$$

Since $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$, we obtain

$$g(\mathbf{x}_K) - g^* \leq \frac{2L_gD^2}{K+1} + \frac{1}{2}\epsilon_g,$$

which completes the proof.

D.2. Proof of Theorem 6

Since we use a fixed stepsize in Theorem 6, in the following we will write $\gamma_k = \gamma$.

We first consider the upper-level objective f. The analysis here is similar to the one in [39]. By using (8) in Lemma 7, we have

$$\mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})}{\gamma} + \frac{1}{2}\gamma L_f D^2.$$

Summing both sides of the above inequality from $k = 0$ to $K - 1$, we get

$$\sum_{k=0}^{K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - f(\mathbf{x}_K)}{\gamma} + \frac{1}{2}K\gamma L_f D^2 \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma} + \frac{1}{2}K\gamma L_f D^2,$$

where we used the fact that $f(\mathbf{x}_K) \geq \underline{f} = \min_{\mathbf{x}\in Z} f(\mathbf{x})$. This further implies that

$$\min_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{1}{K}\sum_{k=0}^{K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma K} + \frac{1}{2}\gamma L_f D^2. \tag{16}$$

To upper bound the right-hand side of (16), note that our choices of the stepsize γ and the number of iterations K satisfy

$$\gamma \leq \frac{\epsilon_f}{L_f D^2} \quad \text{and} \quad K \geq \frac{2(f(\mathbf{x}_0) - \underline{f})}{\epsilon_f \gamma}.$$

Thus, we have

$$\min_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma K} + \frac{1}{2}\gamma L_f D^2 \leq \frac{\epsilon_f}{2} + \frac{\epsilon_f}{2} = \epsilon_f.$$

This guarantees that $\mathcal{G}(\mathbf{x}_{k^*}) \leq \epsilon_f$ by choosing $k^* = \operatorname{argmin}_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k)$.

Now we move to the analysis of the lower-level objective g. For any $k \geq 0$, by applying induction on (9) in Lemma 7 it follows that

$$g(\mathbf{x}_k) - g(\mathbf{x}_0) \leq \frac{1}{2}L_g D^2 \sum_{j=0}^{k-1} \gamma^2(1-\gamma)^j \leq \frac{1}{2}L_g D^2 \gamma,$$

where we used $\sum_{j=0}^{k-1}(1-\gamma)^j \leq 1/\gamma$ in the last inequality. Furthermore, since $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$ and $\gamma \leq \frac{\epsilon_g}{L_g D^2}$, this implies that $g(\mathbf{x}_k) - g^* \leq \frac{1}{2}\epsilon_g + \frac{1}{2}\epsilon_g = \epsilon_g$ for any $0 \leq k \leq K-1$. In particular, we can take $k = k^*$ and conclude that $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$. This completes the proof.

Appendix E. Convergence under Hölderian error bound assumption

In Theorems 4 and 6, we measure the progress for the upper-level objective in terms of $f(\mathbf{x}) - f^*$ (in the convex case) or $\mathcal{G}(\mathbf{x})$ (in the non-convex case). However, in general they may not serve as a good performance metric: since the generated iterate $\mathbf{x}$ may lie outside of the feasible set $\mathcal{X}_g^*$, both $f(\mathbf{x}) - f^*$ and $\mathcal{G}(\mathbf{x})$ could be negative. Thus, our convergence result will be stronger if we can instead upper bound $|f(\mathbf{x}) - f^*|$ or $|\mathcal{G}(\mathbf{x})|$.

Let $\hat{\mathbf{x}}$ be an (ϵ_f, ϵ_g)-optimal solution as defined in Definition 1. Intuitively, since $\hat{\mathbf{x}}$ is ϵ_g-optimal for the lower-level function, it should be close to the optimal solution set $\mathcal{X}_g^*$ under some regularity condition on g. As such, we can lower bound $f(\hat{\mathbf{x}}) - f^*$ by using the smoothness of f. Formally, we assume that the lower-level function satisfies the Hölderian error bound, which quantifies the growth rate of the objective value $g(\mathbf{x})$ as the point $\mathbf{x}$ deviates from the optimal solution set $\mathcal{X}_g^*$.

Assumption 2 *The function g satisfies the Hölderian error bound for some $\alpha > 0$ and $r \geq 1$, i.e,*

$$\frac{\alpha}{r}\ \mathrm{dist}(\mathbf{x}, \mathcal{X}_g^*)^r \leq g(\mathbf{x}) - g^*, \qquad \forall \mathbf{x} \in \mathcal{Z}, \tag{17}$$

where $\mathrm{dist}(\mathbf{x}, \mathcal{X}_g^*) \triangleq \inf_{\mathbf{x}' \in \mathcal{X}_g^*} \|\mathbf{x} - \mathbf{x}'\|$.

We note that the error bound condition in (17) is well-studied in the optimization literature (see [5, 40, 43] and the references therein) and is known to hold generally when the function g is analytic and the set $\mathcal{Z}$ is bounded [35, 36]. Two important special cases are: 1) g satisfies (17) with $r = 1$, i.e., $\mathcal{X}_g^*$ is a set of weak sharp minima of g [7, 8]; 2) g satisfies (17) with $r = 2$, which can be viewed as a general notion of strong convexity.

Under Assumption 2, we can establish the following lower bounds on $f(\hat{\mathbf{x}}) - f^*$ and $\mathcal{G}(\hat{\mathbf{x}})$. Notably, the following result is an intrinsic property of Problem (1) and independent of the algorithm we use.

Proposition 8 *Assume that g satisfies the Hölderian error bound in Assumption 2, and define $M = \max_{\mathbf{x} \in \mathcal{X}_g^*} \|\nabla f(\mathbf{x})\|_*$. Then for any $\hat{\mathbf{x}}$ that satisfies $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$, it holds that:*

(i) If f is convex, then $f(\hat{\mathbf{x}}) - f^* \geq -M \left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}}$.

(ii) If f is non-convex and has L_f-Lipschitz gradient, then $\mathcal{G}(\hat{\mathbf{x}}) \geq -M \left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f \left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}}$.

By combining Theorems 4 and 6 with Proposition 8, we obtain the following stronger convergence guarantees for the output of our proposed method.

Corollary 9 *Suppose that Assumption 1 holds and g satisfies the Hölderian error bound in Assumption 2 with $\alpha > 0$ and $r \geq 1$. Let $M = \max_{\mathbf{x} \in \mathcal{X}_g^*} \|\nabla f(\mathbf{x})\|_*$.*

(i) If f in Problem (1) is convex, we can set $\epsilon_g = \frac{\alpha}{r}\left(\frac{\epsilon_f}{M}\right)^r$. Then after $K = \mathcal{O}(1/\epsilon_f^r)$ iterations, we have $|f(\mathbf{x}_K) - f^*| \leq \epsilon_f$ and $g(\mathbf{x}_K) - g^* \leq \epsilon_g$.

(ii) If f in Problem (1) is non-convex, we can set $\epsilon_g = \min\{\frac{\alpha}{r}\left(\frac{\epsilon_f}{2M}\right)^r, \frac{\alpha}{r}\left(\frac{\epsilon_f}{2L_f}\right)^{r/2}\}$. Then after $K = \mathcal{O}(1/\epsilon_f^{r+1})$ iterations, there exists $k^* \in \{0, 1, \ldots, K-1\}$ such that $|\mathcal{G}(\mathbf{x}_{k^*})| \leq \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$.

Corollary 9 shows that under the r-th Hölderian error bound assumption, we can find an iterate to be ϵ_f-close to optimality within $\mathcal{O}(1/\epsilon_f^r)$ iterations in the convex case, and to be ϵ_f-close to stationarity within $\mathcal{O}(1/\epsilon_f^{r+1})$ iterations in the non-convex case.

E.1. Proof of Proposition 8

Since $\mathcal{X}_g^*$ is closed and compact, we can let $\hat{\mathbf{x}}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}_g^*} \|\mathbf{x} - \hat{\mathbf{x}}\|$ such that $\|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\| = \operatorname{dist}(\hat{\mathbf{x}}, \mathcal{X}_g^*)$. By Assumption 2, we obtain

$$\frac{\alpha}{r}\|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\|^r \le g(\hat{\mathbf{x}}) - g^* \le \epsilon_g \quad \Leftrightarrow \quad \|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\| \le \left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}}.$$

When f is convex, we have

$$f(\hat{\mathbf{x}}) - f^* = f(\hat{\mathbf{x}}) - f(\hat{\mathbf{x}}^*) \ge \langle \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \ge -\|\nabla f(\hat{\mathbf{x}}^*)\|_* \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \ge -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}},$$

where we used the convexity of f in the first inequality. When f is non-convex, we have

$$\begin{aligned}
\mathcal{G}(\hat{\mathbf{x}}) = \max_{\mathbf{s}\in\mathcal{X}_g^*}\{\langle \nabla f(\hat{\mathbf{x}}), \hat{\mathbf{x}} - \mathbf{s}\rangle\} &\ge \langle \nabla f(\hat{\mathbf{x}}), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \\
&= \langle \nabla f(\hat{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle + \langle \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \\
&\ge -\|\nabla f(\hat{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^*)\|_* \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| - \|\nabla f(\hat{\mathbf{x}}^*)\| \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \\
&\ge -L_f\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\|^2 - M\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \qquad (18) \\
&\ge -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}},
\end{aligned}$$

where we used the fact that ∇f is L_f-Lipschitz in (18). This completes the proof.

E.2. Proof of Corollary 9

In the first case where f is convex, we set $\epsilon_g = \frac{\alpha}{r}\left(\frac{\epsilon_f}{M}\right)^r$. By Theorem 4, we have $f(\mathbf{x}_K) - f^* \le \epsilon_f$ and $g(\mathbf{x}_K) - g^* \le \epsilon_g$ when

$$K \ge \max\left\{\frac{2L_f D^2}{\epsilon_f} - 1, \frac{4L_g D^2}{\epsilon_g} - 1\right\} = \max\left\{\frac{2L_f D^2}{\epsilon_f} - 1, \frac{4rM^r L_g D^2}{\alpha\epsilon_f^r} - 1\right\} = \mathcal{O}\left(\frac{1}{\epsilon_f^r}\right).$$

Moreover, Proposition 8 implies that $f(\mathbf{x}_K) - f^* \ge -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} \ge -\epsilon_f$. Putting all pieces together, we conclude that $|f(\mathbf{x}_K) - f^*| \le \epsilon_f$ and $g(\mathbf{x}_K) - g^* \le \epsilon_g$ after $K = \mathcal{O}(1/\epsilon_f^r)$ iterations.

In the second case where f is non-convex, we set $\epsilon_g = \min\{\frac{\alpha}{r}\left(\frac{\epsilon_f}{2M}\right)^r, \frac{\alpha}{r}\left(\frac{\epsilon_f}{2L_f}\right)^{r/2}\}$. By Theorem 6, we can find $k^* \in \{0, 1, \ldots, K-1\}$ such that $\mathcal{G}(\mathbf{x}_{k^*}) \le \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \le \epsilon_g$ when

$$\begin{aligned}
K &\ge (f(\mathbf{x}_0) - \underline{f}) \cdot \max\left\{\frac{2L_f D^2}{\epsilon_f^2}, \frac{2L_g D^2}{\epsilon_f \epsilon_g}\right\} \\
&= (f(\mathbf{x}_0) - \underline{f}) \cdot \max\left\{\frac{2L_f D^2}{\epsilon_f^2}, \frac{2r(2M)^r L_g D^2}{\alpha\epsilon_f^{r+1}}, \frac{2r(2L_f)^{\frac{r}{2}} L_g D^2}{\alpha\epsilon_f^{\frac{r}{2}+1}}\right\} = \mathcal{O}\left(\frac{1}{\epsilon_f^{r+1}}\right).
\end{aligned}$$

Moreover, Proposition 8 implies that $\mathcal{G}(\mathbf{x}_{k^*}) \ge -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}} \ge -\frac{\epsilon_f}{2} - \frac{\epsilon_f}{2} = -\epsilon_f$. Thus, we conclude $|\mathcal{G}(\mathbf{x}_{k^*})| \le \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \le \epsilon_g$ after $K = \mathcal{O}(1/\epsilon_f^{r+1})$ iterations.

Appendix F. Primal-dual method for the bilevel problem

In this section, we discuss the convergence rate of primal-dual type methods for solving the bilevel problem in (1). We consider the setting as in Theorem 4, in which both f and g are convex and smooth. To simplify the discussion, we further assume $\mathcal{Z} = \{\mathbf{z} \in \mathcal{X} \mid \mathbf{A}\mathbf{z} \leq \mathbf{b}\}$ where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathcal{X}$ is a convex and easy-to-project compact set.

Formally, we first reformulate (1) as the following constrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{Z},\, g(\mathbf{x}) \leq g^*. \tag{19}$$

To solve Problem 19, one first needs to estimate the optimal value g^* of the lower-level problem. Since it is a convex program with linear constraints, we can implement a first-order primal-dual method (e.g., [10]) to find g_0 such that $|g_0 - g^*| \leq \epsilon_g/4$ within at most $\mathcal{O}(\frac{L_g+\|\mathbf{A}\|}{\epsilon_g})$ iterations[1]. Next, Problem (1) can be cast as the following convex optimization problem with linear and nonlinear convex constraints:

$$\min_{\mathbf{x}\in\mathcal{X}} \; f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b},\, g(\mathbf{x}) \leq g_0 + \frac{\epsilon_g}{2}, \tag{20}$$

where we add the term $\frac{\epsilon_g}{2}$ to ensure that the Slater's condition holds. Now we can apply any classic or accelerated first-order primal-dual methods [21, 22, 51] to find a solution of Problem (20) that is both ϵ_f-suboptimal and $\frac{\epsilon_g}{4}$-infeasible. For example, the optimal convergence rates obtained in [51] and [21] imply that after K iterations, the average iterate $\bar{\mathbf{x}}_K$ satisfies

$$\max\left\{|f(\bar{\mathbf{x}}_K) - f(\mathbf{x}^*_\epsilon)|\,, |g(\bar{\mathbf{x}}_K) - g(\mathbf{x}^*_\epsilon)|\right\} \leq \Delta/K,$$

where $\mathbf{x}^*_\epsilon$ denotes an optimal solution of Problem (20), $\Delta \triangleq \mathcal{O}((L_f + L_g + C_g)D^2 + C_g\,|\lambda_1^*|^2 + \|\mathbf{A}\|\,\|\boldsymbol{\lambda}_2^*\|^2)$, C_g is the Lipschtiz constant of g, and $\lambda_1^* \in \mathbb{R}$ and $\boldsymbol{\lambda}_2^* \in \mathbb{R}^m$ denote an arbitrary dual optimal solution corresponding to the nonlinear and linear constraints in Problem (20), respectively. Using the fact that $f(\mathbf{x}^*_\epsilon) \leq f(\mathbf{x}^*)$ and $g(\mathbf{x}^*_\epsilon) \leq g_0 + \frac{\epsilon_g}{2} \leq g^* + \frac{3}{4}\epsilon_g$, we conclude

$$f(\bar{\mathbf{x}}_K) - f(\mathbf{x}^*) \leq \Delta/K \quad \text{and} \quad |g(\bar{\mathbf{x}}_K) - g(\mathbf{x}^*)| \leq \Delta/K + \frac{3}{4}\epsilon_g.$$

Therefore, to achieve an (ϵ_f, ϵ_g)-optimal solution of Problem (1), a primal-dual method overall requires $\mathcal{O}\left(\frac{L_g+\|\mathbf{A}\|}{\epsilon_g} + \frac{\Delta}{\min\{\epsilon_f,\epsilon_g\}}\right)$ primal-dual gradient calls, while our proposed method overall requires $\mathcal{O}\left(\frac{L_g}{\epsilon_g} + \frac{(L_f+L_g)D^2}{\min\{\epsilon_f,\epsilon_g\}}\right)$ linear minimization oracle calls. In particular, we observe that the convergence guarantee of primal-dual methods heavily rely on the norm of the dual optimal variable $|\lambda_1^*|$, which may tend to infinity as ϵ approaches zero and the problem in (20) becomes nearly degenerate.

Appendix G. Numerical experiments

In this section, we test our method for solving different bilevel optimization problems. First, we consider a toy example to demonstrate the instability of primal-dual methods by comparing the iteration trajectory of our method with accelerated primal-dual method with backtracking (APDB) proposed by [21]. Next, we consider the hyperparameter selection problem described in Example 2 and compare our method with other existing methods in the literature [3, 26, 45]. All experiments are performed on a MacBook Pro with Apple M1 chip and 16GB RAM.

1. Note that this complexity can be improved to the optimal rate of $\mathcal{O}(\sqrt{\frac{L_g}{\epsilon_g}} + \frac{\|\mathbf{A}\|}{\epsilon_g})$ using an accelerated method.

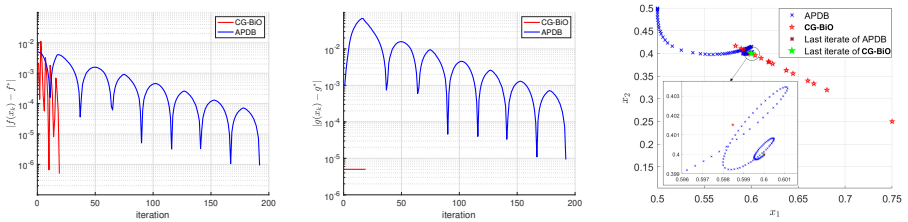


Figure 1: The performance of CG-BiO (red) vs APDB (blue) on Problem (21). Plots from left to right: upper-level suboptimality, lower-level suboptimality, and iteration trajectory.

G.1. Toy example

Here we consider a simple two-dimensional example to illustrate the numerical instability of primal-dual methods applied to the relaxed problem (20). To this end, consider the following problem

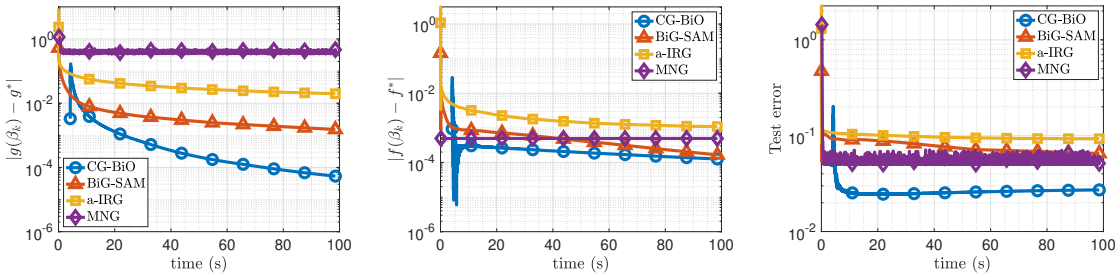
$$\min_{\mathbf{x}\in\mathbb{R}^2} 0.5x_1^2 - 0.5x_1 + 0.1x_2 \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}}\{-z_1 - z_2\}, \tag{21}$$

where $\mathcal{Z} = \{\mathbf{z} \in \mathbb{R}_+^n \mid z_1 + z_2 \leq 1, 4z_1 + 6z_2 \leq 5\}$. The lower-level problem has multiple solutions which can be described by $\mathcal{X}_g^* = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1 + x_2 = 1, x_1 \in [0.5, 1], x_2 \in [0, 0.5]\}$ and the optimal solution of (21) is $(x_1^*, x_2^*) = (0.6, 0.4)$. We implemented our proposed method and compared it with APDB. Figure 1 illustrates the iteration trajectories of both methods. We selected the relaxing parameter in (20) as $\epsilon = 10^{-5}$ for APDB. We also used the same accuracy for ϵ_g and ϵ_f when implementing CG-BiO. The primal-dual method finds an ϵ-solution (dark red cross) within 193 iterations while CG-BiO finds an ϵ-solution (green star) within 20 iterations. Furthermore, we observe a more stable numerical behavior for CG-BiO in comparison with APDB. This is consistent with our discussions in Appendix F.

G.2. Hyperparameter selection

In this section, we consider a sparse linear regression problem on the Wikipedia Math Essential dataset [44], which consists of a data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $n = 1068$ instances and $d = 730$ attributes and an outcome vector $\mathbf{b} \in \mathbb{R}^n$. Our goal is to find a sparse parameter $\boldsymbol{\beta} \in \mathbb{R}^d$ to achieve a small prediction error $\frac{1}{2}\|\mathbf{A}\boldsymbol{\beta} - \mathbf{b}\|_2^2$. We formulate the regression problem as the bilevel optimization problem in Example 2. Specifically, we assign 60% of the dataset as the training set $(\mathbf{A}_{\text{tr}}, \mathbf{b}_{\text{tr}})$, 20% as the validation set $(\mathbf{A}_{\text{val}}, \mathbf{b}_{\text{val}})$ and the rest as the test set $(\mathbf{A}_{\text{test}}, \mathbf{b}_{\text{test}})$. Then the lower-level objective is the training error $g(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{A}_{\text{tr}}\mathbf{z} - \mathbf{b}_{\text{tr}}\|_2^2$, the upper-level objective is the validation error $f(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{A}_{\text{val}}\boldsymbol{\beta} - \mathbf{b}_{\text{val}}\|_2^2$, and the constraint set is the ℓ_1-ball $\mathcal{Z} = \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_1 \leq \lambda\}$ for some $\lambda > 0$ to induce sparsity in $\boldsymbol{\beta}$. We also use the test error $\frac{1}{2}\|\mathbf{A}_{\text{test}}\boldsymbol{\beta} - \mathbf{b}_{\text{test}}\|_2^2$ as our performance metric. Note that the regression problem is over-parameterized since the number of features d is larger than the number of data instances in the training set.

In the experiment, we implement our CG-BiO algorithm to solve the bilevel problem with parameter $\lambda = 1$. We set the target accuracies for the upper-level and lower-level problems to $\epsilon_f = 10^{-4}$ and $\epsilon_g = 10^{-4}$, respectively. For comparison, we also implement the MNG method in [3], the Bilevel Gradient SAM (BiG-SAM) in [45], and the averaging iteratively regularized gradient (a-IRG) method in [26]. For benchmarking purposes, we use CVX [19, 20] to solve the



(*a*) Lower-level gap (*b*) Upper-level gap (*c*) Test error objective

Figure 2: The performance of CG-BiO compared with BiG-SAM, a-IRG and MNG on hyperparameter selection. Plots from left to right: lower-level suboptimality, upper-level suboptimality, and the test error.

lower-level problem and the constrained reformulation in (19) to obtain the optimal values g^* and f^*, respectively.

In Fig. 2, we illustrate the numerical performance of all considered algorithms. From Fig. 2(a), we can see that CG-BiO converges at a faster rate than the other baseline methods in terms of the lower-level objective, which confirms our theoretical result (cf. Table 1). Fig. 2(c) and (d) also show that it is able to achieve a smaller upper-level objective value as well as a smaller test error compared with BiG-SAM, a-IRG and MNG within the same running time. Interestingly, we observe that after the initial stage, the upper-level objective $f(\boldsymbol{\beta}_k)$ of CG-BiO actually *increases*, while the optimality gap $|f(\boldsymbol{\beta}_k) - f^*|$ *decreases*. This suggests that CG-BiO may "overshoot" at the beginning due to its relatively large stepsize. Nevertheless, as the number of iterations increases and the level of infeasibility decreases, the upper-level objective of our algorithm approaches the optimal value of the bilevel problem, which is also in line with Proposition 8.

Appendix H. Further experiment details

In this section, we include more details of the numerical experiments in Section G.

For completeness, we briefly review the update rules of MNG [3], BiG-SAM [45], and a-IRG [26] in the setup of Problem (1). In the following, we use $\Pi_{\mathcal{Z}}(\cdot)$ to denote the Euclidean projection onto the set $\mathcal{Z}$.

- Each step of MNG requires solving the following subproblem:

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in Q_k \cap W_k}{\operatorname{argmin}} \; f(\mathbf{x}), \tag{22}$$

where

$$Q_k \triangleq \left\{ \mathbf{z} \in \mathbb{R}^n : \langle G_M(\mathbf{x}_k), \mathbf{x}_k - \mathbf{z} \rangle \geq \frac{3}{4M} \|G_M(\mathbf{x}_k)\|^2 \right\},$$
$$W_k \triangleq \{ \mathbf{z} \in \mathbb{R}^n : \langle \nabla f(\mathbf{x}_k), \mathbf{z} - \mathbf{x}_k \rangle \geq 0 \},$$
$$G_M(\mathbf{x}) \triangleq M \left[\mathbf{x} - \Pi_{\mathcal{Z}} \left(\mathbf{x} - \frac{1}{M} \nabla g(\mathbf{x}) \right) \right],$$

and $M \leq 1/L_g$ is a hyperparameter. As we can see, the implementation of MNG is only feasible when the subproblem in (22) is easy to solve. In particular, it is computationally intractable when the upper-level objective f is non-convex.

- BiG-SAM is given by

$$\mathbf{y}_{k+1} = \Pi_{\mathcal{Z}}(\mathbf{x}_k - \eta_g \nabla g(\mathbf{x}_k)), \tag{23}$$
$$\mathbf{z}_{k+1} = \mathbf{x}_k - \eta_f \nabla f(\mathbf{x}_k), \tag{24}$$
$$\mathbf{x}_{k+1} = \alpha_{k+1}\mathbf{z}_{k+1} + (1-\alpha_{k+1})\mathbf{y}_{k+1}, \tag{25}$$

  where η_f and η_g are stepsizes and $\alpha_k = \min\{\frac{\gamma}{k}, 1\}$ for some $\gamma > 0$.

- The a-IRG algorithm is given by

$$\mathbf{x}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{x}_k - \gamma_k(\nabla g(\mathbf{x}_k) + \eta_k \nabla f(\mathbf{x}_k))\right), \tag{26}$$

  where γ_k is the stepsize and η_k is the regularization parameter. In our experiment, we choose $\gamma_k = 0.01/\sqrt{k+1}$ and $\eta_k = 1/(k+1)^{1/4}$.

H.1. Over-parametrized regression

Dataset generation. The original Wikipedia Math Essential dataset [44] consists of an 1068×731 matrix. We randomly select one of the columns as the outcome vector $\mathbf{b} \in \mathbb{R}^{1068}$ and the rest as the data matrix $\mathbf{A} \in \mathbb{R}^{1068\times 730}$.
Initialization. We run the standard CG algorithm with the stepsize chosen as $2/(k+2)$ on the lower-level problem. We terminate the procedure once the FW gap is no more than $\epsilon_g/2 = 5 \times 10^{-5}$.
Implementation details. For our CG-BiO algorithm, we choose the stepsizes as $\gamma_k = 2/(k+12)$ to avoid instability due to large stepsizes. In each iteration, we need to solve a subproblem in the form of

$$\min_{\mathbf{s}} \ \langle \nabla f(\boldsymbol{\beta}_k), \mathbf{s}\rangle \qquad \text{s.t.} \quad \|\mathbf{s}\|_1 \leq \lambda, \ \langle \nabla g(\boldsymbol{\beta}_k), \mathbf{s} - \boldsymbol{\beta}_k\rangle \leq g(\boldsymbol{\beta}_0) - g(\boldsymbol{\beta}_k). \tag{27}$$

We can reformulate the above problem as a linear program by introducing $\mathbf{s}^+, \mathbf{s}^- \geq 0$ such that $\mathbf{s} = \mathbf{s}^+ - \mathbf{s}^-$. Specifically, Problem (27) becomes

$$\begin{aligned}
&\min_{\mathbf{s}^+,\mathbf{s}^-} \ \langle \nabla f(\boldsymbol{\beta}_k), \mathbf{s}^+ - \mathbf{s}^-\rangle \\
&\text{s.t.} \quad \mathbf{s}^+, \mathbf{s}^- \geq 0, \ \langle \mathbf{s}^+, \mathbf{1}\rangle + \langle \mathbf{s}^-, \mathbf{1}\rangle \leq \lambda, \ \langle \nabla g(\boldsymbol{\beta}_k), \mathbf{s}^+ - \mathbf{s}^- - \boldsymbol{\beta}_k\rangle \leq g(\boldsymbol{\beta}_0) - g(\boldsymbol{\beta}_k).
\end{aligned}$$