Noname manuscript No.
(will be inserted by the editor)

Direct Search meets Frank-Wolfe: Inexact Derivative-Free Bilevel Optimization over Polytopes

Matteo Pernini · Francesco Rinaldi · Gabriele Sanguin · Marco Viola

Received: date / Accepted: date

Abstract In this work we propose BLDS-SIMPLEX, a derivative-free direct search method for the solution of bilevel optimization with polytope constraints at the upper level. The proposed method reformulates the feasible set using barycentric coordinates and explores it using Frank-Wolfe-type directions, relying on a fixed-accuracy black-box oracle for the upper-level objective function. We analyze its convergence properties and complexity in the case of smooth reduced objective; additionally, we analyze a variant for the case of nonsmooth objectives and show its convergence to a specific kind of stationary point that we introduce as a natural generalization to the classic stationarity notions in this setting. Numerical tests on modified instances from BOLIB support the viability of our new approach.

Keywords Bilevel optimization · Derivative-free Optimization · Direct search · Frank Wolfe · Inexact lower level

M. Pernini
ESSEC Business School, Paris (France)
E-mail: matteo.pernini@essec.edu

F. Rinaldi, G. Sanguin
Department of Mathematics "Tullio Levi-Civita", University of Padova, Padova (Italy)
E-mail: rinaldi@math.unipd.it, gabriele.sanguin.2@phd.unipd.it

M. Viola
School of Mathematical Sciences, Dublin City University, Dublin (Ireland)
E-mail: marco.viola@dcu.ie

1 Introduction

We consider bilevel optimization problems with polytope constraints at the upper level:

$$\begin{aligned} \min_{x \in \Omega_x} \quad & f(x, y^*(x)) \\ \text{s.t.} \quad & y^*(x) \in \arg\min_{y \in \Omega_y} g(x, y), \end{aligned} \tag{1}$$

where $\Omega_x \subseteq \mathbb{R}^{n_x}$ is a nonempty polytope and $\Omega_y \subseteq \mathbb{R}^{n_y}$ is the lower-level feasible set. We focus on problems with a unique lower-level response and access to approximate lower-level solutions computed at a fixed accuracy. The upper-level algorithm uses the resulting inexact objective evaluations without requiring derivative information. The precise assumptions on the problem and the oracle are stated in Section 3. Our goal is to study the convergence properties of a derivative-free optimization (DFO) method in both smooth and nonsmooth settings.

Contributions. In this work, we make the following contributions:

1. **Algorithm design.** We propose BLDS-SIMPLEX (Algorithm 1), a derivative-free direct-search algorithm for bilevel optimization on polytopes. The method reformulates the upper-level problem on the unit simplex through an atomic representation of the feasible set and explores Frank-Wolfe-type directions using inexact upper-level function evaluations obtained through a lower-level oracle.
2. **Convergence in the smooth setting.** Under suitable smoothness assumptions and bounded function-evaluation errors, we establish finite termination when a positive minimum stepsize is enforced. More specifically, we derive a bound on the Frank-Wolfe gap at the final iterate, quantifying the effects of oracle accuracy and algorithmic parameters (Theorem 1).
3. **Analysis in the nonsmooth setting.** We introduce the notion of a Nearby-Clarke Frank-Wolfe stationary point (NCFWSP) and study convergence properties of a suitable variant of the method using this criterion (Theorem 2). The analysis builds on the inexact direct-search framework of [14], accounting for oracle inexactness and polytope constraints at the upper level.
4. **Numerical experiments.** We evaluate the proposed method on modified bilevel benchmark problems from the BOLIB library [33,36] and compare its empirical performance with that of existing derivative-free approaches.

Paper outline. The rest of the paper is organized as follows. Section 2 reviews the related literature. Section 3 introduces the problem formulation, the algorithm, and some necessary assumptions and definitions. Section 4 presents the convergence analysis in both the smooth and nonsmooth settings. Section 5 reports the numerical results. Section 6 concludes the paper.

2 Related Work

Bilevel optimization. Bilevel optimization has a long history in operations research and mathematical programming [9,13]. Interest in this topic has grown significantly in recent years due to applications in machine learning, such as hyperparameter optimization and meta-learning (see, e.g., [17,18] and references therein). For differentiable convex lower-level problems satisfying suitable constraint qualifications, classical approaches replace the lower-level optimality requirement with its KKT conditions, yielding an equivalent mathematical program with equilibrium constraints (MPEC) [9]. More recently, gradient-based methods that exploit explicit or implicit differentiation to compute hypergradients have gained popularity in machine learning (see, e.g., [35] and references therein for a survey). Many implicit-differentiation approaches require suitable differentiability assumptions and access to derivative information. These derivative requirements can limit applicability in black-box settings, where such information may be unavailable or costly to obtain. Moreover, hypergradient calculation can be computationally expensive and sensitive to the conditioning of the lower-level problem. For these reasons, our work studies a derivative-free method that only requires inexact function evaluations of the upper-level objective, making it applicable to a broader class of bilevel problems with black-box structure.

Derivative-free methods. Derivative-free optimization (DFO) methods are designed to minimize functions without requiring or using derivative information. They can be broadly categorized into direct-search, model-based and randomized methods (see, e.g., [2,10,15,22] and references therein for further details on these three classes of methods).

In the context of bilevel optimization, several derivative-free approaches have been developed. To cite some, Mersha and Dempe [27] and Zhang and Lin [34] applied direct-search methods to bilevel problems with various assumptions on the lower-level oracle, respectively assuming no approximation error and adaptive error. Conn and Vicente [11] developed a trust-region framework accommodating inexact lower-level solutions and reuse of sample points. Ehrhardt and Roberts [16] analyzed a dynamically accurate derivative-free trust-region method and established worst-case complexity bounds. More recently, zeroth-order methods based on smoothing strategies [7,25] have been studied, offering tailored complexity estimates. Finally, min-max DFO problems [1,26] and direct-search methods under adaptive deterministic noise [24] and stochastic noise [4,30] have also been tackled in the literature.

Most closely related to our work is the analysis of Diouane et al. [14], who developed directional and mesh-based direct-search methods for bilevel problems with a fixed-accuracy lower-level oracle. They analyzed convergence in both smooth and nonsmooth settings and provided complexity estimates. Their analysis, however, concerns unconstrained problems. Our work builds on this framework by considering polytope constraints at the upper level while retaining the fixed-accuracy oracle model.

To the best of our knowledge, previous analyses of direct-search methods for bilevel optimization have not simultaneously considered bounded, non-adaptive lower-level inexactness and polytope constraints at the upper level. Our approach differs from existing methods by explicitly exploiting the polytope geometry through Frank-Wolfe-type directions. Consequently, its convergence guarantees are expressed in terms of Frank-Wolfe gap-based stationarity measures. In this respect, our approach shares some similarities with the simplicial direct-search framework introduced by Cristofari and Rinaldi [12]. However, their scheme exploits barycentric coordinates and sparse feasible directions to guarantee convergence for single-level black-box problems, while we extend these concepts to an inexact bilevel setting where approximate lower-level solutions lead to inexact upper-level evaluations.

In a recent work, Cesaroni et al. [6] proposed a derivative-free framework for bilevel optimization that accounts for lower-level inexactness. Their framework differs from ours in two main respects. First, our algorithm is a feasible method designed to directly handle upper-level feasible regions given by general polytopes, whereas Cesaroni et al. focus primarily on bound-constrained problems and treat general nonlinear upper-level constraints via an exact-penalty reformulation. Second, their analysis is built around a progressive refinement scheme for lower-level accuracy, establishing convergence to Clarke-Jahn stationary points as the lower-level error tends to zero and approximate Goldstein stationarity when refinement is limited, whereas our approach directly analyzes a fixed lower-level inexactness.

3 Preliminaries

3.1 Problem Formulation

We now address the general formulation of the bilevel optimization problem (1), with a nonempty compact upper-level feasible set Ω_x. We will later specialize to the case where Ω_x is a polytope described as the convex hull of a finite set of atoms. First, notice that there might be different solutions to the lower-level problem, which can lead to different upper-level objectives. To ensure that the upper-level objective is well-defined and to simplify the analysis, it is common practice in the literature to assume that the lower-level problem has a unique solution for each x, which is known as the lower-level singleton (LLS) assumption (see, e.g., [14] and references therein for further details).

Assumption 1 (LLS)**.** *For any $x \in \Omega_x$, we have that* $\arg\min_{y\in\Omega_y} g(x,y) = \{y^*(x)\}$.

Under this assumption, problem (1) reduces to

$$\min_{x\in\Omega_x} F(x) := f(x, y^*(x)). \tag{2}$$

In practical implementations, it is often not possible to compute the lower-level solution $y^*(x)$ exactly. Instead, we assume to have access to an approximate

solution $\tilde{y}(x)$ leading to the inexact objective $\tilde{F}(x) := f(x, \tilde{y}(x))$. By calling this *lower-level oracle*, the algorithm always operates on $\tilde{F}$, bringing the need for the following two assumptions to analyze its convergence in terms of the true objective F:

Assumption 2 (Bounded Lower-Level Error)**.** *For any $x \in \Omega_x$, the approximate lower-level solution $\tilde{y}(x) \in \Omega_y$ satisfies $\|\tilde{y}(x) - y^*(x)\| \leq \varepsilon$ for a given $\varepsilon > 0$.*

Assumption 3 (Lipschitz Continuity of f w.r.t. y)**.** *The function $f(x, y)$ is L_f-Lipschitz continuous with respect to y, uniformly in x:*

$$|f(x, y_1) - f(x, y_2)| \leq L_f \|y_1 - y_2\|, \quad \forall\, y_1, y_2 \in \Omega_y,\ \forall\, x \in \Omega_x.$$

Assumption 2 says that the lower-level oracle always gives an approximate solution within a given error bound ε from the exact solution. We remark that this can be satisfied by applying standard solvers to the lower-level problem, e.g. gradient descent or DFO methods with suitable error bounds (see discussion in [14]). With Assumption 3, we are giving a bound on the change in the upper-level objective function due to changes in the lower-level solution. By combining the two, we get an error bound between the inexact and exact objectives:

Remark 1 (Bounded objective error) Assumptions 2 and 3 imply:

$$|\tilde{F}(x) - F(x)| \leq L_f \varepsilon \text{ for all } x \in \Omega_x$$

.

To study convergence both in the smooth and nonsmooth case, we also need the following continuity assumption on the upper-level objective function F.

Assumption 4 (Lipschitz Continuity of F)**.** *$F(x)$ is L_F-Lipschitz continuous in Ω_x, i.e.*

$$|F(x_1) - F(x_2)| \leq L_F \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \Omega_x.$$

Note that if f is Lipschitz continuous with respect to x, and $y^*(x)$ is Lipschitz continuous with respect to x, then Assumption 4 is satisfied.

Remark 2 (Lower boundedness of F) Under Assumption 4, the objective function $F(x)$ is bounded below on Ω_x by some constant $f_{\min} > -\infty$, due to the compactness of the domain (we assume Ω_x closed and bounded) and the continuity of F. By the error bound of Remark 1, we also have that

$$\tilde{F}(x) \geq \tilde{f}_{\min} := f_{\min} - L_f \varepsilon > -\infty \text{ for all } x \in \Omega_x.$$

As anticipated above, from now on we are considering the upper-level feasible set $\Omega_x \subseteq \mathbb{R}^{n_x}$ to be a polytope described by a finite set of *atoms* $\mathcal{A} = \{a_1, \dots, a_m\} \subset \mathbb{R}^{n_x}$, so that $\Omega_x = \text{conv}(\mathcal{A})$. Any point $x \in \Omega_x$ can therefore be described as a convex combination $x = Aw$, where $A = [a_1\ a_2\ \cdots\ a_m] \in \mathbb{R}^{n_x \times m}$ is the atom matrix and $w \in \Delta^{m-1} := \{w \in \mathbb{R}^m : w \geq 0,\ \mathbf{1}^\top w = 1\}$ is the vector of barycentric coordinates. Thus, problem (1) can be equivalently reformulated considering the simplicial representation of the feasible set:

$$\begin{aligned} \min_{w \in \Delta^{m-1}} \ & f(Aw, y^*(Aw)) \\ \text{s.t. } & y^*(Aw) \in \arg\min_{y \in \Omega_y} g(Aw, y). \end{aligned} \tag{3}$$

Equivalently, we can write the reduced problem (2) as

$$\min_{w \in \Delta^{m-1}} F_A(w) := f(Aw, y^*(Aw)). \tag{4}$$

The assumptions above transfer to the simplex formulation through the map $x = Aw$. In particular, the inexact objective is

$$\tilde{F}_A(w) := \tilde{F}(Aw) = f(Aw, \tilde{y}(Aw)).$$

By Remark 1, the objective-error bound is preserved:

$$|\tilde{F}_A(w) - F_A(w)| \leq L_f \varepsilon, \qquad \forall w \in \Delta^{m-1}.$$

Moreover, Assumption 4 gives

$$|F_A(w_1) - F_A(w_2)| \leq L_F \|A\| \, \|w_1 - w_2\|, \qquad \forall w_1, w_2 \in \Delta^{m-1}.$$

Thus, $L_{F_A} := L_F \|A\|$ is a valid Lipschitz constant for F_A. Here vector norms are Euclidean and $\|A\|$ denotes the induced operator norm.

3.2 Algorithm

The proposed algorithm, BLDS-SIMPLEX, is a *BiLevel Direct-Search method on the Unit Simplex* that iteratively explores Frank-Wolfe-type directions $d = \pm(e_i - w^k)$, where e_i is the i-th standard basis vector. At each iteration, the algorithm cycles through a permutation of the coordinate indices and, for each index, tests both positive and negative directions.

The pseudocode of the proposed method is reported in Algorithm 1. We recall the definition of the maximum stepsize that guarantees feasibility:

Definition 1 (Maximum Feasible Step) Let $w^k \in \Delta^{m-1}$. We define the maximum feasible step for $d \in \mathbb{R}^m$, as

$$\bar{\alpha}(w^k, d) := \max\{\alpha \geq 0 : w^k + \alpha d \in \Delta^{m-1}\}. \tag{5}$$

Note that for the specific case of FW directions in the simplex, the computation of the maximal feasible stepsize has the closed form:

$$\bar{\alpha}(w^k, d) = \begin{cases} 1 & \text{if } d = e_i - w^k, \\ \frac{w_i^k}{1 - w_i^k} & \text{if } d = -(e_i - w^k). \end{cases}$$

In BLDS-SIMPLEX, a trial point $w^k + \alpha d$ is accepted if it produces a sufficient decrease in the inexact objective:

$$\tilde{F}_A(w^k + \alpha d) \leq \tilde{F}_A(w^k) - \gamma(\alpha^k)^2,$$

where $\gamma > 0$ is a forcing parameter and $\alpha = \min\{\alpha^k, \bar{\alpha}(w^k, d)\}$ is the effective stepsize, truncated at the maximum feasible step $\bar{\alpha}(w^k, d)$ to keep the trial point on the simplex. Upon a successful step, the nominal stepsize is enlarged, subject to the prescribed maximum; if all polling directions fail, it is contracted, subject to the minimum stepsize. When a positive minimum stepsize $\alpha_{\min} > 0$ is enforced, the algorithm terminates after an unsuccessful iteration with $\alpha^k = \alpha_{\min}$, at which point the stepsize can no longer be reduced.

Algorithm 1 BLDS-SIMPLEX

Require: Choose $w^0 \in \Delta^{m-1}$, initial stepsize $\alpha^0 \in (0, 1]$, minimum stepsize $\alpha_{\min} \in [0, 1]$, maximum stepsize $\alpha_{\max} \in (0, 1]$, parameters $\theta \in (0, 1)$, $\delta \in (0, 1)$, and forcing constant $\gamma > 0$

for $k = 0, 1, 2, \ldots$ **do**
  Set `success` = `false` and $\alpha^k = \max\{\alpha^k, \alpha_{\min}\}$
  Choose a permutation of $\{1, \ldots, m\}$ and denote its elements as $\{i_j\}_{j=1}^m$. Set $j = 1$
  while $j \leq m$ **do**
    for sgn $\in \{+, -\}$ **do**
      Set $d = (\text{sgn})(e_{i_j} - w^k)$
      Compute $\bar{\alpha}(w^k, d)$, the largest α such that $w^k + \alpha d \in \Delta^{m-1}$
      Set $\hat{\alpha} = \min\{\alpha^k, \bar{\alpha}(w^k, d)\}$
      if $\hat{\alpha} > 0$ **then**
        Call lower-level oracle to compute $\tilde{y}(A(w^k + \hat{\alpha} d))$ and evaluate $\tilde{F}_A(w^k + \hat{\alpha} d)$
        if $\tilde{F}_A(w^k + \hat{\alpha} d) \leq \tilde{F}_A(w^k) - \gamma(\alpha^k)^2$ **then**
          Set $w^{k+1} = w^k + \hat{\alpha} d$ and `success` = `true`
          break (exit the sign loop)
        end if
      end if
    end for
    if `success` **then**
      break (exit the while loop)
    end if
    Set $j = j + 1$
  end while
  if `success` **then**
    Set $\alpha^{k+1} = \min\{\alpha_{\max}, \alpha^k / \delta\}$ ▷ Expand stepsize
  else
    Set $w^{k+1} = w^k$ and $\alpha^{k+1} = \max\{\alpha_{\min}, \theta \alpha^k\}$ ▷ Shrink stepsize
    if some approximate stationarity condition satisfied **then**
      Terminate ▷ Stopping criterion
    end if
  end if
end for

4 Theoretical Analysis

In this section, we present the convergence analysis of Algorithm 1. We first consider the smooth setting (Section 4.1), where the reduced objective F has a Lipschitz continuous gradient, and then the nonsmooth setting (Section 4.2), where only Lipschitz continuity of F is assumed.

4.1 Convergence in the Smooth Setting

In this subsection, we impose the following smoothness assumption on the reduced objective.

Assumption 5 (Smoothness of F)**.** *The reduced objective $F(x) = f(x, y^*(x))$ admits a continuously differentiable extension to an open neighborhood of Ω_x, denoted again by F. Its gradient is L-Lipschitz continuous on Ω_x, i.e.,*

$$\|\nabla F(x_1) - \nabla F(x_2)\| \leq L\|x_1 - x_2\|, \qquad \forall x_1, x_2 \in \Omega_x.$$

For the simplex reformulation $F_A(w) = F(Aw)$, the chain rule gives

$$\nabla F_A(w) = A^\top \nabla F(Aw).$$

Consequently, F_A has a Lipschitz continuous gradient on Δ^{m-1} with a valid constant

$$L_A := L\|A\|^2.$$

This smoothness assumption allows us to apply the descent lemma to control directional derivatives along the forward Frank–Wolfe directions at unsuccessful iterations. These bounds yield a stationarity guarantee expressed in terms of the Frank–Wolfe gap, which we recall next.

Definition 2 (Frank-Wolfe Gap) For a differentiable function F on a convex set Ω, the Frank-Wolfe gap at $x \in \Omega$ is

$$g_{\mathrm{FW}}(x; F, \Omega) := \max_{u \in \Omega} \langle -\nabla F(x),\, u - x \rangle. \tag{6}$$

A point x^* is ϵ-Frank-Wolfe stationary if $g_{\mathrm{FW}}(x^*; F, \Omega) \leq \epsilon$.

The Frank-Wolfe gap is a common criterion for measuring convergence rate of projection-free algorithms in the smooth setting: it vanishes at first-order stationary points and can be efficiently computed for polytopes, as it reduces to a linear optimization problem over the feasible set.

Let us start now with the convergence analysis of Algorithm 1. To recall everything seen until now, we are considering the reduced bilevel problem (4) under Assumptions 1, 2, 3 and 5, which will be referred as the *smooth setting assumptions*. Note that Assumption 5 directly implies Assumption 4 on the compact domain Ω_x. Our goal is to show convergence to a point with a small Frank-Wolfe gap (2) and give the complexity bounds when $\alpha_{\min} \geq 0$. We begin with a lemma that bounds the steepest feasible *FW-direction* at an unsuccessful iteration.

Lemma 1 (Gradient bound at unsuccessful iterations) *Let the smooth setting assumptions hold, and let* $\mathcal{D}_{\Delta}^{k} := \{e_i - w^k : i = 1, \dots, m\}$ *be the set of elementary forward FW-directions at iteration* k*. If iteration* k *is unsuccessful, then, for every* $d \in \mathcal{D}_{\Delta}^{k}$*,*

$$-\nabla F_A(w^k)^\top d \leq (L_A + \gamma)\alpha^k + \frac{2L_f\varepsilon}{\alpha^k}. \tag{7}$$

Proof. At iteration k, consider any FW-direction $d = e_i - w^k \in \mathcal{D}_{\Delta}^{k}$. If $d = 0$, the claimed inequality holds immediately. Otherwise, the maximum feasible stepsize along d is one. Since $0 < \alpha^k \leq 1$, the algorithm therefore evaluates the feasible trial point $w^k + \alpha^k d$ without truncating the nominal stepsize.

Because the iteration is unsuccessful, we have

$$\tilde{F}_A(w^k + \alpha^k d) > \tilde{F}_A(w^k) - \gamma(\alpha^k)^2. \tag{8}$$

From Remark 1, we have

$$\begin{aligned} \tilde{F}_A(w^k) &\geq F_A(w^k) - L_f\varepsilon, \\ \tilde{F}_A(w^k + \alpha^k d) &\leq F_A(w^k + \alpha^k d) + L_f\varepsilon. \end{aligned}$$

Combining these inequalities with (8), we obtain

$$F_A(w^k + \alpha^k d) - F_A(w^k) > -\gamma(\alpha^k)^2 - 2L_f\varepsilon. \tag{9}$$

By Assumption 5 and the linear reformulation, F_A has an L_A-Lipschitz continuous gradient on Δ^{m-1}. Thus, the descent lemma gives

$$\begin{aligned} F_A(w^k + \alpha^k d) - F_A(w^k) \leq \alpha^k \nabla F_A(w^k)^\top d \\ + \frac{L_A}{2}(\alpha^k)^2\|d\|^2. \end{aligned} \tag{10}$$

Combining (9) and (10), and dividing by $\alpha^k > 0$, we find

$$-\nabla F_A(w^k)^\top d < \left(\gamma + \frac{L_A}{2}\|d\|^2\right)\alpha^k + \frac{2L_f\varepsilon}{\alpha^k}.$$

Finally, since $w^k \in \Delta^{m-1}$, we have $\|d\|^2 \leq 2$. Consequently,

$$-\nabla F_A(w^k)^\top d \leq (L_A + \gamma)\alpha^k + \frac{2L_f\varepsilon}{\alpha^k},$$

which proves the claimed bound. □

We now state the main convergence result for the smooth case.

Theorem 1 (Smooth convergence and FW-gap bound) *Let the smooth setting assumptions hold. Then:*

1. *If $\alpha_{\min} > 0$, Algorithm 1 terminates at an iteration $\bar{k}$ satisfying*

$$\bar{k} \leq 1 + \frac{\tilde{F}_A(w^0) - \tilde{f}_{\min}}{\gamma \alpha_{\min}^2} \left(2 + \frac{\ln \delta}{\ln \theta}\right) + \frac{\ln \alpha_{\min} - \ln \alpha^0}{\ln \theta},$$

and the final iterate $w^{\bar{k}}$ satisfies:

$$g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) \leq \Big(L_A + \gamma\Big)\alpha_{\min} + \frac{2L_f \varepsilon}{\alpha_{\min}}. \tag{11}$$

2. *If $\alpha_{\min} = 0$, then $\alpha^k \to 0$. Moreover, for any $\bar{\alpha} > 0$ with $\alpha^0 \geq \bar{\alpha}$, there exists an unsuccessful iteration $\bar{k}$ such that:*

$$g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) < \frac{1}{\theta}\Big[\Big(L_A + \gamma\Big)\bar{\alpha} + \frac{2L_f \varepsilon}{\bar{\alpha}}\Big], \tag{12}$$

and for all $k \geq \bar{k}$,

$$F_A(w^k) \leq F_A(w^{\bar{k}}) + 2L_f \varepsilon.$$

Proof. Consider the sequence $\{\tilde{F}_A(w^k)\}$. It is nonincreasing by definition and bounded below (see Remark 2); therefore, whenever the algorithm generates an infinite sequence, these values converge to some finite limit $\ell \geq \tilde{f}_{\min}$.

Suppose now $\alpha_{\min} > 0$. It follows that the number k_s of successful iterations is finite:

$$k_s \leq \frac{\tilde{F}_A(w^0) - \tilde{f}_{\min}}{\gamma \alpha_{\min}^2}, \tag{13}$$

where we used the fact that, for each successful step, the decrease in the objective is at least $\gamma \alpha_{\min}^2$. After the last successful iteration, consecutive unsuccessful iterations reduce the stepsize to $\alpha_{\min}$, and the next unsuccessful iteration terminates the algorithm. Thus termination occurs after finitely many iterations.

Let k_{ns} be the number of unsuccessful iterations, so that $k_s + k_{ns} = \bar{k} + 1$ is the total number of iterations. Among the nonterminal unsuccessful iterations, let k_f count those for which $\theta \alpha^k \geq \alpha_{\min}$, and let k_c count those for which $\theta \alpha^k < \alpha_{\min}$.

On a successful iteration, the stepsize is increased by at most a factor $1/\delta$; on each of the k_f iterations, it is multiplied by θ; and the remaining updates cannot increase it. Since the final stepsize is $\alpha_{\min}$, we have

$$\alpha_{\min} \leq \frac{\theta^{k_f}}{\delta^{k_s}} \alpha^0. \tag{14}$$

Moreover, each of the k_c clipped contractions sets the next stepsize to $\alpha_{\min}$. Before another such contraction can occur, there must be a successful iteration; otherwise, the next unsuccessful iteration would terminate the algorithm. Hence $k_c \leq k_s + 1$.

Including the terminal unsuccessful iteration, we have $k_{ns} = k_f + k_c + 1$. Using (14), we can therefore bound k_{ns} as follows:

$$\begin{aligned} k_{ns} &\leq 2 + k_s + \frac{1}{\ln(\theta)} \big(\ln(\alpha_{\min}) - \ln(\alpha^0) + k_s \ln(\delta)\big) \\ &\leq 2 + \frac{\tilde{F}_A(w^0) - \tilde{f}_{\min}}{\gamma \alpha_{\min}^2} \left(1 + \frac{\ln(\delta)}{\ln(\theta)}\right) + \frac{\ln(\alpha_{\min}) - \ln(\alpha^0)}{\ln(\theta)}. \end{aligned} \tag{15}$$

We then combine the bounds on successful and unsuccessful iterations (13) and (15) to obtain the desired bound on $\bar{k}$:

$$\begin{aligned} \bar{k} &= k_s + k_{ns} - 1 \\ &\leq 1 + \frac{1}{\gamma \alpha_{\min}^2} (\tilde{F}_A(w^0) - \tilde{f}_{\min}) \left(2 + \frac{\ln(\delta)}{\ln(\theta)}\right) + \frac{\ln(\alpha_{\min}) - \ln(\alpha^0)}{\ln(\theta)}. \end{aligned} \tag{16}$$

To prove the bound on the FW-gap, we use Lemma 1, which holds for every unsuccessful iteration. From the algorithm stopping criterion, the last iteration $\bar{k}$ is unsuccessful with $\alpha^{\bar{k}} = \alpha_{\min}$. Therefore, for every forward FW-direction $d \in \mathcal{D}_{\Delta}^{\bar{k}}$, Equation (7) gives:

$$-\nabla F_A(w^{\bar{k}})^\top d \;\leq\; \Big(L_A + \gamma\Big)\, \alpha_{\min} \;+\; \frac{2 L_f\, \varepsilon}{\alpha_{\min}}.$$

Suppose now there exists at least one point $\hat{u} \in \Delta^{m-1}$ such that

$$\big\langle -\nabla F_A(w^{\bar{k}}),\, \hat{u} - w^{\bar{k}} \big\rangle > 0.$$

If no such point exists, we immediately obtain

$$g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) = 0,$$

and the claimed bound follows. Otherwise,

$$\begin{aligned} g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) &:= \max_{u \in \Delta^{m-1}} \big\langle -\nabla F_A(w^{\bar{k}}),\, u - w^{\bar{k}} \big\rangle \\ &= \max_{u \in \Delta^{m-1}} \Big\langle -\nabla F_A(w^{\bar{k}}),\, \sum_{i=1}^{m} u_i\, (e_i - w^{\bar{k}}) \Big\rangle \\ &= \max_{u \in \Delta^{m-1}} \sum_{i=1}^{m} u_i \big\langle -\nabla F_A(w^{\bar{k}}),\, e_i - w^{\bar{k}} \big\rangle \\ &\leq \max_{u \in \Delta^{m-1}} \sum_{i=1}^{m} u_i \max_{j \in \{1,\ldots,m\}} \big\langle -\nabla F_A(w^{\bar{k}}),\, e_j - w^{\bar{k}} \big\rangle \\ &\leq \max_{j \in \{1,\ldots,m\}} \big\langle -\nabla F_A(w^{\bar{k}}),\, e_j - w^{\bar{k}} \big\rangle \\ &\leq \max_{d \in \mathcal{D}_{\Delta}^{\bar{k}}} \big(-\nabla F_A(w^{\bar{k}})^\top d\big). \end{aligned}$$

Combining the previous inequalities, we obtain the FW-gap bound:

$$g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) \;\leq\; \Big(L_A + \gamma\Big)\,\alpha_{\min} \;+\; \frac{2L_f\,\varepsilon}{\alpha_{\min}}.$$

This concludes the proof for the case $\alpha_{\min} > 0$.

Let us now consider the case $\alpha_{\min} = 0$. In this case, the algorithm does not terminate in a finite number of iterations. Summing the sufficient decreases over the successful iterations gives

$$\gamma \sum_{k \in \mathcal{S}} (\alpha^k)^2 \leq \tilde{F}_A(w^0) - \tilde{f}_{\min},$$

where $\mathcal{S}$ is the set of successful iteration indices. Hence, for every $\tau > 0$, there are only finitely many successful iterations with $\alpha^k \geq \delta\tau$. Beyond the last such iteration, every successful iteration produces $\alpha^{k+1} < \tau$, while unsuccessful iterations contract the stepsize. Consequently, the stepsize eventually falls below τ and remains below it. Since $\tau > 0$ is arbitrary, $\alpha^k \to 0$.

Assuming $\alpha^0 \geq \bar{\alpha}$, let $j \geq 1$ be the first index such that $\alpha^j < \theta\bar{\alpha}$. Since successful iterations cannot decrease the stepsize, iteration $j-1$ is unsuccessful and $\alpha^j = \theta\alpha^{j-1}$. Setting $\bar{k} = j-1$, we therefore have $\alpha^{\bar{k}} \in [\theta\bar{\alpha}, \bar{\alpha})$.

Define now $\phi(\alpha) = a\alpha + b/\alpha$, with $a = L_A + \gamma$ and $b = 2L_f\varepsilon$; ϕ is continuous and convex on $\mathbb{R}_{>0}$. Note that, for $\theta \in (0,1)$,

$$\phi(\theta\bar{\alpha}) < \frac{1}{\theta}\phi(\bar{\alpha}) \qquad \text{and} \qquad \phi(\bar{\alpha}) < \frac{1}{\theta}\phi(\bar{\alpha}).$$

Therefore, $\phi(\alpha^{\bar{k}}) \leq \max\{\phi(\bar{\alpha}), \phi(\theta\bar{\alpha})\} < \frac{1}{\theta}\phi(\bar{\alpha})$. Applying Lemma 1 at the unsuccessful iteration $\bar{k}$ and using the simplex representation of the FW-gap above gives Equation (12).

Finally, for any $\bar{k} \in \mathbb{N}_0$ and $k \geq \bar{k}$, we have:

$$F_A(w^k) \leq \tilde{F}_A(w^k) + L_f\varepsilon \leq \tilde{F}_A(w^{\bar{k}}) + L_f\varepsilon \leq F_A(w^{\bar{k}}) + 2L_f\varepsilon.$$

□

Corollary 1 (Optimal stepsize choice) *The FW-gap bound in* (11) *is minimized by choosing*

$$\alpha^*_{\min} = \sqrt{\frac{2L_f\varepsilon}{L_A + \gamma}},$$

provided $L_f\varepsilon > 0$ *and* $\alpha^*_{\min} \leq \alpha^0$. *This choice yields:*

$$g_{\mathrm{FW}}(w^{\bar{k}}; F_A, \Delta^{m-1}) \leq 2\sqrt{2L_f\varepsilon\Big(L_A + \gamma\Big)} = O(\sqrt{\varepsilon}),$$

as $\varepsilon \downarrow 0$ *with the other constants fixed.*

4.2 Convergence in the Nonsmooth Setting

In this subsection, we relax Assumption 5 and require only Lipschitz continuity of the reduced objective F, together with Assumptions 1–4. This case leads to a nonsmooth polytope-constrained upper-level problem in which we do not have access to an exact lower-level solution. To adapt to this scenario, we need to make some changes to the original BLDS-SIMPLEX algorithm and to introduce an appropriate measure of stationarity.

Without differentiability, the Frank-Wolfe gap is no longer well-defined, because of its dependence on the gradient. Therefore, we introduce the concept of (δ, ϵ)-Nearby-Clarke Frank-Wolfe Stationary Point (NCFWSP), aimed at extending the Frank-Wolfe gap to nonsmooth objectives through the Clarke subdifferential. We first recall some definitions.

Definition 3 (Clarke Subdifferential [8]) Let $F : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz at x. The Clarke subdifferential of F at x is

$$\partial F(x) := \{g \in \mathbb{R}^n : DF(x; v) \geq \langle g, v\rangle \ \ \forall\, v \in \mathbb{R}^n\},$$

where $DF(x; v) := \limsup_{y \to x,\, t \downarrow 0} \frac{F(y+tv)-F(y)}{t}$ is the generalized directional derivative. Each element g of $\partial F(x)$ is called a generalized gradient of F at x.

Definition 4 (Clarke Frank-Wolfe gap [23]) Let $\Omega \subseteq \mathbb{R}^n$ be nonempty, compact, and convex, and let F be locally Lipschitz on an open neighborhood of Ω. For $x \in \Omega$, the Clarke Frank-Wolfe gap at x is

$$g_{\mathrm{FW}}^{\mathrm{Cl}}(x;\, F,\, \Omega) \ := \ \min_{g \in \partial F(x)} \ \max_{u \in \Omega} \langle -g,\, u - x\rangle.$$

We are now ready to introduce the concept of Nearby-Clarke Frank-Wolfe Stationarity.

Definition 5 (Nearby-Clarke Frank-Wolfe Stationary Point (NCFWSP)) Under the same assumptions on F and Ω, a point $x \in \Omega$ is a (δ, ϵ)-NCFWSP if

$$\exists \tilde{x} \in \mathbb{B}_\delta(x) \cap \Omega \quad \text{s.t.} \quad g_{\mathrm{FW}}^{\mathrm{Cl}}(\tilde{x};\, F,\, \Omega) \leq \epsilon.$$

When F is continuously differentiable, the Clarke Frank-Wolfe gap coincides with the classical Frank-Wolfe gap. The nearby-point definition therefore reduces to approximate smooth stationarity at a nearby feasible point, and to the usual condition at x when $\delta = 0$.

Remark 3 An alternative stationarity definition for the nonsmooth constrained case, namely the Goldstein Frank-Wolfe Stationary Point (GFWSP), can be found in [23] as a point satisfying

$$\min_{g \in \partial_\delta F(x)} \ \max_{u \in \Omega} \langle -g,\, u - x\rangle,$$

where, given $\delta > 0$, the gap is measured by considering the δ-Goldstein subdifferential [19], defined as

$$\partial_\delta F(x) := \operatorname{conv}\left(\bigcup_{y \in \mathbb{B}_\delta(x)} \partial F(y) \right).$$

In simpler words, in [23] the gap is measured at the point x itself using an element of the Goldstein subdifferential, which includes subgradients at nearby points. In our opinion, this might be excessively restrictive while the definition we give in this paper is conceptually more suitable for a constrained setting and is a more direct generalization of the Clarke Frank-Wolfe (CFW) gap. In fact, in this paper, for a point to be stationary it suffices that it is in the δ-neighborhood of a point $\tilde{x}$ that is CFW stationary.

Since we are working with Clarke subdifferentials, the analysis in this subsection relies on a variant of Algorithm 1 in which the finite set of Frank-Wolfe directions $\mathcal{D}_\Delta^k = \{\pm(e_i - w^k)\}_{i=1}^m$ is replaced by a *dense* set of directions in the tangent cone to the simplex at w^k. In practice, only Lines 3-6 of Algorithm 1 change: instead of cycling over atoms, the algorithm draws directions from a dense subset of cone$\{e_i - w^k\}_{i=1}^m$. Crucially, the maximum feasible stepsize $\bar{\alpha}(w^k, d)$ still admits a closed-form expression for any direction d in the tangent cone to the simplex.

Assumption 6 (Dense Direction Set)**.** *During every iteration k, the set of directions considered by the algorithm at lines 3-6 is dense in the cone generated by the elementary Frank-Wolfe directions $\{e_i - w^k\}_{i=1}^m$, denoted as cone$(\mathcal{D}_\Delta^k)$.*

We first establish that, with $\alpha_{\min} > 0$, the iterates become eventually constant.

Lemma 2 *Under Assumptions 1-4, with $\gamma > 0$ and $\alpha_{\min} > 0$, the sequence $\{w^k\}$ produced by Algorithm 1 is eventually constant: there exists $\bar{k}$ such that $w^{k+1} = w^k$ for all $k \geq \bar{k}$.*

Proof. Every successful iteration decreases $\tilde{F}_A$ by at least $\gamma\alpha_{\min}^2$. Since $\tilde{F}_A$ is bounded below by $\tilde{f}_{\min}$, the number of successful iterations is finite. After the last successful iteration, all subsequent iterations are unsuccessful and the iterate remains unchanged. □

Theorem 2 (Nonsmooth convergence to NCFWSP) *Let Assumptions 1–4 and 6 hold, with $\gamma > 0$ and $\alpha_{\min} > 0$. Define $\eta := 2^{\frac{3}{2}}\big(\gamma\alpha_{\min} + 2L_f\varepsilon/\alpha_{\min}\big)$. Then every limit point w^* of the sequence $\{w^k\}$ is an $(\alpha_{\min}, \eta)$-Nearby-Clarke Frank-Wolfe stationary point for F_A on Δ^{m-1}: there exists $\tilde{w}^* \in \mathbb{B}_{\alpha_{\min}}(w^*) \cap \Delta^{m-1}$ such that*

$$\min_{g \in \partial F_A(\tilde{w}^*)} \max_{u \in \Delta^{m-1}} \langle -g,\, u - \tilde{w}^* \rangle \leq 2^{\frac{3}{2}}\left(\gamma\alpha_{\min} + \frac{2L_f\varepsilon}{\alpha_{\min}}\right). \tag{17}$$

Proof. By Lemma 2 the sequence $\{w^k\}$ is eventually constant and we denote its limit value with w^*. For k large enough, every direction trial is unsuccessful and $\alpha^k = \alpha_{\min}$, therefore for every $d \in \mathit{cone}(\mathcal{D}^*_\Delta)$ the effective stepsize used by the algorithm is:

$$\hat{\alpha}^*(d) := \min\{\alpha_{\min}, \bar{\alpha}(w^*, d)\}.$$

By the unsuccessful condition, we have:

$$\tilde{F}_A(w^* + \hat{\alpha}^*(d)d) \;>\; \tilde{F}_A(w^*) - \gamma\alpha^2_{\min}.$$

Using Remark 1 (bounded approximation error $|\tilde{F}_A - F_A| \leq L_f\varepsilon$) we obtain, for every $d \in \mathit{cone}(\mathcal{D}^*_\Delta)$,

$$F_A(w^*) \;<\; F_A(w^* + \hat{\alpha}^*(d)d) + \gamma\alpha^2_{\min} + 2L_f\varepsilon. \tag{18}$$

We call the tangent cone to the simplex at a point w as $T_{\Delta^{m-1}}(w)$. Define the set

$$\mathcal{H} := \{h = \alpha_{\min} d : d \in T_{\Delta^{m-1}}(w^*),\ \|d\| = 1\}$$

of tangent displacements with norm $\alpha_{\min}$, and the set

$$\mathcal{G} := \{g = \hat{\alpha}^*(d)d : d \in T_{\Delta^{m-1}}(w^*),\ \|d\| = 1\}$$

of clipped displacements for which $w^* + g$ is feasible. Let $\chi_\Omega(x)$ be the characteristic function of the set Ω, i.e. $\chi_\Omega(x) = 0$ if $x \in \Omega$ and $+\infty$ otherwise. From (18) and density of the generated directions it holds that, for every $h \in \mathcal{H}$,

$$F_A(w^*) \;<\; F_A(w^* + h) \;+\; C \;+\; \chi_{\Delta^{\bar{m}-1}}(w^* + h), \qquad C := \gamma\alpha^2_{\min} + 2L_f\varepsilon,$$

where the penalty $\chi_{\Delta^{\bar{m}-1}}(w^* + h) = +\infty$ is given when $w^* + h$ is not feasible.

Let us define the auxiliary function:

$$\bar{F}_x(h) \;:=\; F_A(x + h) \;+\; C\frac{\|h\|^2}{\alpha^2_{\min}} \;+\; \chi_{\Delta^{\bar{m}-1}}(w^* + h),$$

and note that

$$\bar{F}_{w^*}(0) < \bar{F}_{w^*}(h) \quad \text{for every } h \in \mathcal{H}. \tag{19}$$

We now consider the compact sets bounded by $\mathcal{H}$ and $\mathcal{G}$ and centered at the origin, namely:

$$\bar{\mathcal{H}} := \{\tau h \mid h \in \mathcal{H}, \tau \in [0,1]\} \subseteq \mathbb{B}_{\alpha_{\min}}(0),$$
$$\bar{\mathcal{G}} := \{\tau h \mid h \in \mathcal{G}, \tau \in [0,1]\} \subseteq \bar{\mathcal{H}}.$$

Figure 1 provides a diagram which can help understand the difference between the four different sets involved here. Notice that $\bar{F}_x(h) = +\infty$ for every $h \in \bar{\mathcal{H}} \backslash \bar{\mathcal{G}}$, and that $\{w^* + h \mid h \in \bar{\mathcal{G}}\} \subseteq \Delta^{\bar{m}-1}$.

By compactness of $\bar{\mathcal{G}}$ and the continuity of F_A, there exists a minimizer $\tilde{h} \in \bar{\mathcal{G}}$ of $\bar{F}_{w^*}$ which is also a minimizer inside $\bar{\mathcal{H}}$. Since the boundary of $\bar{\mathcal{H}}$ is suboptimal from (19), thus $\|\tilde{h}\| < \alpha_{\min}$, i.e. the minimizer is an interior

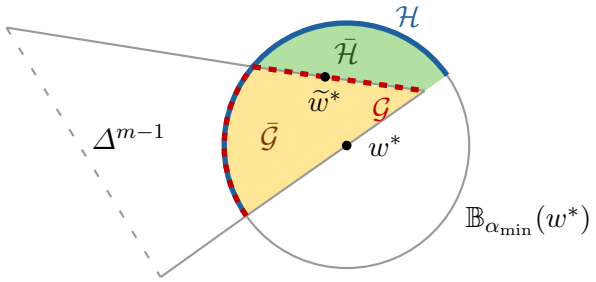


Fig. 1: Example of sets $\mathcal{H}$ (blue), $\mathcal{G}$ (red dashed), $\bar{\mathcal{H}}$ (green, including the golden region), and $\bar{\mathcal{G}}$ (golden) from the proof of Theorem 2. With a slight abuse of notation, all four sets are shown translated by the vector w^*. Both shaded regions include their boundaries. The golden region is the intersection of the closed tangent half-disk with the simplex; in particular, its straight boundary segments are included in $\bar{\mathcal{G}}$, even where they do not belong to $\mathcal{G}$.

point of $\bar{\mathcal{H}}$ and $w^* + \tilde{h} \in \Delta^{\bar{m}-1}$ because $\tilde{h} \in \bar{\mathcal{G}}$. It holds the following Clarke Frank-Wolfe optimality condition:

$$\min_{g \in \partial F_{w^*}(\tilde{h})} \max_{d \in T_{\bar{\mathcal{G}}}(w^* + \tilde{h})} \langle -g, d \rangle = 0. \tag{20}$$

Moreover, it holds that:

$$T_{\bar{\mathcal{G}}}(\tilde{h}) = T_{\Delta^{\bar{m}-1}}(w^* + \tilde{h}) \tag{21}$$

because $\{w^* + \mathcal{G}\} \subset \Delta^{\bar{m}-1}$ and if $\tilde{h}$ is in the border of $\mathcal{G}$, then $w^* + \tilde{h}$ is in the border of $\Delta^{\bar{m}-1}$.

Therefore, from (20) and (21), and the fact that $\partial F_{w^*}(\tilde{h}) = \partial F_A(w^* + \tilde{h}) + \frac{2C}{\alpha^2_{\min}}\tilde{h}$, it follows that:

$$\min_{g \in \partial F_A(w^* + \tilde{h})} \max_{u \in \Delta^{\bar{m}-1}} \langle -g - \tfrac{2C}{\alpha^2_{\min}}\tilde{h}, u - (w^* + \tilde{h}) \rangle = 0 \tag{22}$$

$$\Rightarrow \min_{g \in \partial F_A(w^* + \tilde{h})} \max_{u \in \Delta^{\bar{m}-1}} \left(\langle -g, u - (w^* + \tilde{h}) \rangle - \frac{2C}{\alpha^2_{\min}} \langle \tilde{h}, u - (w^* + \tilde{h}) \rangle \right) = 0. \tag{23}$$

Now, we know that for general functions $A(x)$ and $B(x)$, we have $\max_x (A(x) - B(x)) = 0 \Rightarrow A(x) \leq B(x) \quad \forall x$; in particular $\max_x A(x) \leq B(\arg\max_x (A(x))) \leq \max_x B(x)$. Using this property, from (23) we imply that there exists $\bar{g} \in \partial F_A(w^* + \tilde{h})$ such that:

$$\max_{u \in \Delta^{\bar{m}-1}} \left(\langle -\bar{g}, u - (w^* + \tilde{h}) \rangle - \frac{2C}{\alpha^2_{\min}} \langle \tilde{h}, u - (w^* + \tilde{h}) \rangle \right) = 0$$

$$\Rightarrow \max_{u \in \Delta^{\bar{m}-1}} \langle -\bar{g}, u - (w^* + \tilde{h}) \rangle \leq \max_{u \in \Delta^{\bar{m}-1}} \frac{2C}{\alpha^2_{\min}} \langle \tilde{h}, u - (w^* + \tilde{h}) \rangle \leq \frac{2C}{\alpha^2_{\min}} \|\tilde{h}\| D.$$

Finally, using $D = \sqrt{2}$, it follows that:

$$\min_{g \in \partial F_A(w^* + \tilde{h})} \max_{u \in \Delta^{\bar{m}-1}} \langle -g, u - (w^* + \tilde{h}) \rangle \leq \frac{2C}{\alpha_{\min}^2} \|\tilde{h}\| D$$

$$\Rightarrow \exists \tilde{w}^* = w^* + \tilde{h} \in \mathbb{B}_{\alpha_{\min}}(w^*) \text{ s.t. } \min_{g \in \partial F_A(\tilde{w}^*)} \max_{u \in \Delta^{m-1}} \langle -g, u - \tilde{w}^* \rangle \leq \frac{2^{\frac{3}{2}} C}{\alpha_{\min}}.$$

□

5 Numerical Experiments

In this section we present our results validating BLDS-SIMPLEX (Algorithm 1) on bilevel benchmark problems from the BOLIB library [36,33], showing that BLDS-SIMPLEX is always able to find high-quality solutions. As for number of function evaluations, we note that in worst case scenarios it must check $O(m)$ directions at each iteration. To make the algorithm more scalable and competitive, we also analyze the performance of a variant in which we take a random subset of FW directions at each iteration, instead of the full batch. The variant searches randomly ordered blocks of Frank–Wolfe directions and stops the search as soon as a sufficient-decrease step is found. This subspace variant BLDS-SIMPLEX-S, described in Algorithm 2, reduces in fact the computational cost per iteration without sacrificing solution quality, making it a promising choice in practice. The results are compared with well known solvers for both bilevel and single-level derivative-free optimization. Experiments can be reproduced with the code available in GitHub [1].

Test problems and polytope constraint construction. We select nine bilevel problems with dimensions n_x ranging from 2 to 10 and n_y from 2 to 12, covering linear, quadratic, and nonlinear objectives; see Table 1. For each problem, the upper-level polytope $\Omega_x = \text{conv}(a_1, \ldots, a_m)$ is constructed by sampling m vertices uniformly on the surface of a ball of radius $r = 5$ in $\mathbb{R}^{n_x}$. The polytope is then shifted so that the known optimal solution x^* is positioned at the centroid.

Each of the 9 problems is combined with $m \in \{50, 100, 200, 500, 1000\}$ atoms and 3 random seeds ($\{11, 22, 33\}$), for a total of $9 \times 5 \times 3 = 135$ instances per solver (27 per fixed m). In the discussion we group vertex counts into *small* instances ($m \in \{50, 100, 200\}$) and *large* instances ($m \in \{500, 1000\}$) to highlight scalability trends.

Lower-level oracle. For each upper-level trial point x, the lower-level problem is solved by SciPy's SLSQP [21,32] with a fixed tolerance of 10^{-3}. We stress that no rigorous certificate of attaining a prescribed accuracy ε is available at every call; SLSQP is used as a practical black-box oracle with a fixed stopping criterion.

[1] https://github.com/GibboSan/df-blo

Table 1: Test problems from BOLIB [33,36] Objective types at upper level (UL) and lower level (LL): L = linear, Q = quadratic, NL = nonlinear.

Problem	n_x	n_y	**UL**	**LL**
DempeLohse2011Ex31a	2	2	Q	L
CalamaiVicente1994b	4	2	Q	Q
CalamaiVicente1994c	4	2	Q	Q
SinhaMaloDeb2014TP9	10	10	Q	NL
SinhaMaloDeb2014TP10	10	10	Q	NL
NieWangYe2017Ex54	4	4	NL	Q
hypercube_10	10	10	L	L
qbp_8_12	8	12	Q	Q
qbp_9_9	9	9	Q	Q

Algorithm 2 BLDS-SIMPLEX-S

Require: $w^0 \in \Delta^{m-1}$, α^0, $\alpha_{\min}$, $\alpha_{\max}$, $\theta, \delta \in (0,1)$, $\gamma > 0$, block size $s \geq 1$
for $k = 0, 1, 2, \ldots$ **do**
  `success` $\leftarrow$ `false`; $\alpha \leftarrow \max\{\alpha^k, \alpha_{\min}\}$
  Randomly partition $\{1, \ldots, m\}$ into blocks $\mathcal{B}_1, \ldots, \mathcal{B}_{\lceil m/s \rceil}$
  for $b = 1, \ldots, \lceil m/s \rceil$ **do**
    for each $i \in \mathcal{B}_b$ and sgn $\in \{+, -\}$ **do**
      $d \leftarrow \text{sgn}\,(e_i - w^k)$; $\hat{\alpha} \leftarrow \min\{\alpha, \bar{\alpha}(w^k, d)\}$
      if $\hat{\alpha} > 0$ **and** $\tilde{F}_A(w^k + \hat{\alpha} d) \leq \tilde{F}_A(w^k) - \gamma(\alpha)^2$ **then**
        $w^{k+1} \leftarrow w^k + \hat{\alpha} d$; `success` $\leftarrow$ `true`; **break**
      end if
    end for
    if `success` **then** $\alpha \leftarrow \min\{\alpha_{\max}, \alpha/\delta\}$; **break**
    else $\alpha \leftarrow \max\{\alpha_{\min}, \theta\alpha\}$
    end if
  end for
  $\alpha^{k+1} \leftarrow \alpha$.
  if not `success` and $\alpha^{k+1} = \alpha^k$ **then Terminate.**
  end if
end for

Subspace variant. Cycling through all $2m$ Frank-Wolfe directions per iteration becomes expensive. Algorithm 2 mitigates this by randomly partitioning $\{1, \ldots, m\}$ into blocks of size s and searching one block at a time: upon finding a sufficient-decrease step the iteration ends immediately; otherwise the stepsize is contracted and the next block is tried. We set $s = n_x$, keeping the per-block cost proportional to the original problem dimension. When $s = m$ the algorithm reduces to the original. The same convergence guarantees hold. In the experiment we will report the subspace variant as **BLDS** and the original strategy as **BLDS (full)**.

Competing algorithms and parameters. We compare BLDS-SIMPLEX and BLDS-SIMPLEX-S against five derivative-free solvers, all using the simplex formulation in (4) and the same lower-level oracle: **DFN-LLA** [6] (based on an exact penalty function), **DSPFD** [20] (using probabilistic feasible directions, with the original MATLAB implementation available here), **LINCOA** [28] (a model-based trust-region method from the PDFO library [29]), **NOMAD** [3,5]

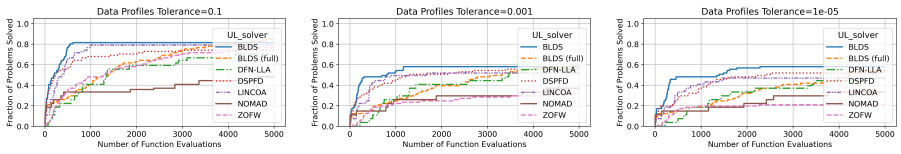
(a mesh adaptive direct-search method from PyNomadBBO), and **ZOFW** [31] (zeroth-order Frank-Wolfe with I-RDSA gradient estimation).

The single-level solvers DSPFD, LINCOA, NOMAD, and ZOFW are applied directly to the inexact reduced objective $\tilde{F}_A(w) = \tilde{F}(Aw) = f(Aw, \tilde{y}(Aw))$ over the simplex. These solvers do not explicitly exploit the bilevel structure: they treat the values returned by the common lower-level oracle as black-box objective evaluations. Thus, they operate as single-level methods optimizing an inexact objective, whose evaluation errors arise from the approximate solution of the lower-level problem.

All solvers share a budget of $N_{\max} = 5000$ inexact objective evaluations. For our algorithms, we set $\alpha^0 = 0.4$, $\alpha_{\min} = 10^{-5}$, $\alpha_{\max} = 0.8$, $\theta = \delta = 0.5$, and $\gamma = 10^{-3}$. The initial point is a randomly generated interior point of the polytope, with a fixed seed for reproducibility.

Metrics. Performance is assessed via *data profiles*: the fraction of problems solved as a function of the number of function evaluations, where a problem is solved when $f^k \leq f^* + \epsilon_{\text{tol}}$. We report results for $\epsilon_{\text{tol}} \in \{10^{-1}, 10^{-3}, 10^{-5}\}$.

Results. We first report data profiles on the whole benchmark set (all values of m aggregated), at three tolerances; see Figure 2.



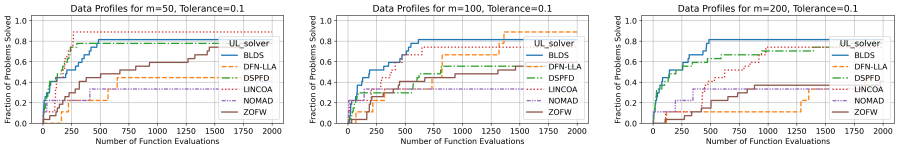
(a) $\epsilon_{\text{tol}} = 10^{-1}$ (b) $\epsilon_{\text{tol}} = 10^{-3}$ (c) $\epsilon_{\text{tol}} = 10^{-5}$

Fig. 2: Data profiles on the whole benchmark set.

From the plots it is clear how the proposed BLDS (subspace variant) is able to outperform all the competitors in terms of efficiency and robustness. The plots also highlight the advantage of the subspace variant over the "full" version, whose search over the complete set of FW directions can increase both the per-iteration cost and the total number of function evaluations. Based on these results, we omit "BLDS (full)" from the subsequent plots.

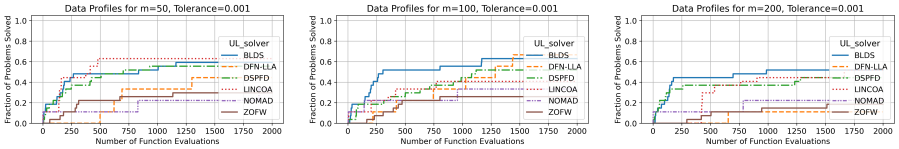
For small instances ($m \in \{50, 100, 200\}$), Figures 3–5 show the behavior at the three tolerances.

From the plots one can clearly observe that while LINCOA and DSPFD show better performances in the case of smaller instances ($m = 50$), as the problem size increases ($m \in \{100, 200\}$) BLDS becomes the most efficient method in all accuracy settings. It is worth mentioning that, despite its lower efficiency, DFN-LLA shows good robustness for the case $m = 100$, while the proposed BLDS is the most robust method in the case $m = 200$ in all accuracy settings.



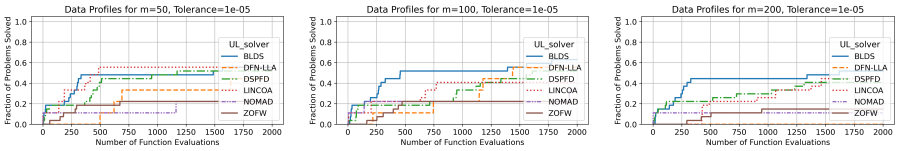
(a) $m = 50$ (b) $m = 100$ (c) $m = 200$

Fig. 3: Data profiles at $\epsilon_{\text{tol}} = 10^{-1}$ for small vertex counts.



(a) $m = 50$ (b) $m = 100$ (c) $m = 200$

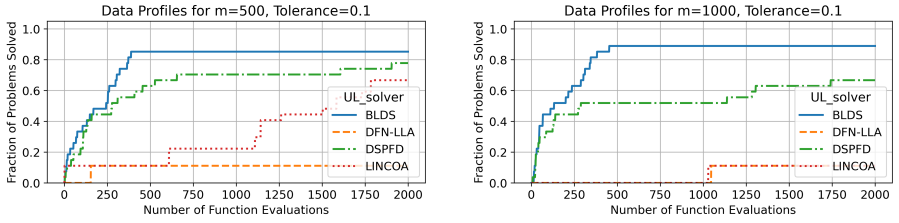
Fig. 4: Data profiles at $\epsilon_{\text{tol}} = 10^{-3}$ for small vertex counts.



(a) $m = 50$ (b) $m = 100$ (c) $m = 200$

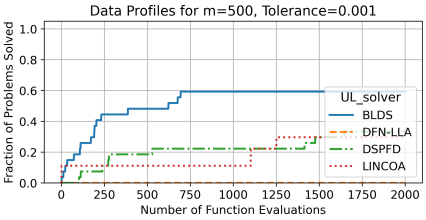
Fig. 5: Data profiles at $\epsilon_{\text{tol}} = 10^{-5}$ for small vertex counts.

In Figures 6–8 we report the results obtained for the largest instances considered, namely $m \in \{500, 1000\}$. For these plots we decided to remove NOMAD and ZOFW as they struggled with these problem sizes.

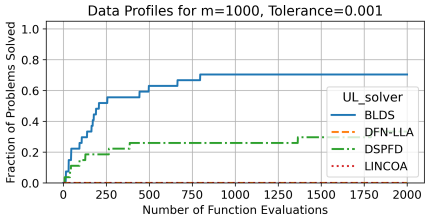


(a) $m = 500$ (b) $m = 1000$

Fig. 6: Data profiles at $\epsilon_{\text{tol}} = 10^{-1}$ for large vertex counts.

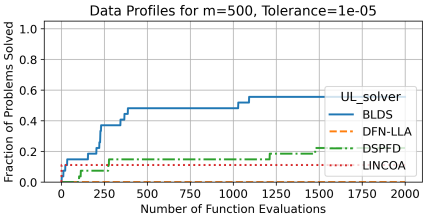


(a) $m = 500$

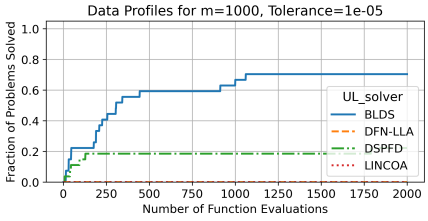


(b) $m = 1000$

Fig. 7: Data profiles at $\epsilon_{\text{tol}} = 10^{-3}$ for large vertex counts.



(a) $m = 500$



(b) $m = 1000$

Fig. 8: Data profiles at $\epsilon_{\text{tol}} = 10^{-5}$ for large vertex counts.

The results highlight the scalability advantage of BLDS, which for all problem sizes and accuracy settings is the most efficient and robust strategy among the four considered ones.

6 Conclusion

We have proposed a derivative-free direct-search algorithm for bilevel optimization problems where the upper-level feasible set is a polytope. By reformulating the problem on the unit simplex via barycentric coordinates, the method exploits Frank-Wolfe-type directions to maintain feasibility while requiring only inexact zeroth-order evaluations from a lower-level oracle.

The main contribution of our work is the theoretical analysis, which establishes convergence guarantees in two settings. In the smooth case, we proved finite termination for a positive minimum stepsize and bounded the Frank-Wolfe gap at the final iterate, showing that a suitable choice of the minimum stepsize yields an $O(\sqrt{\varepsilon})$ gap. In the nonsmooth case, we introduced the notion of a Nearby-Clarke Frank-Wolfe stationary point (NCFWSP) and established approximate stationarity guarantees for a dense-direction variant under Lipschitz continuity of the reduced objective, accounting for both the minimum stepsize and the oracle error. These results extend the analysis of inexact direct-search methods to the constrained bilevel setting through Frank-Wolfe gap-based stationarity measures. We also performed numerical experiments to assess

the performance of the proposed strategy against established derivative-free solvers applied to a suitable reformulation of the bilevel problem. The results illustrate the competitiveness of the proposed strategy on the tested instances, particularly for larger numbers of vertices and more stringent target accuracies.

In future work, we aim to analyze extensions of the proposed strategy to polyhedral feasible sets specified by linear constraints, including cases where the full list of vertices is not known a priori or is expensive to compute. We also plan to investigate computationally feasible direction-sampling strategies for the dense-direction variant used in the nonsmooth analysis.

References

1. Anagnostidis, S.K., Lucchi, A., Diouane, Y.: Direct-search for a class of stochastic min-max problems. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, *Proceedings of Machine Learning Research*, vol. 130, pp. 3772–3780. PMLR (2021). URL `https://proceedings.mlr.press/v130/anagnostidis21a.html`
2. Audet, C.: A survey on direct search methods for blackbox optimization and their applications. In: P.M. Pardalos, T.M. Rassias (eds.) Mathematics Without Boundaries: Surveys in Interdisciplinary Research, pp. 31–56. Springer (2014). doi:10.1007/978-1-4939-1124-0_2
3. Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization **17**(1), 188–217 (2006). doi:10.1137/040603371
4. Audet, C., Dzahini, K.J., Kokkolaras, M., Le Digabel, S.: Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates. Computational Optimization and Applications **79**(1), 1–34 (2021). doi:10.1007/s10589-020-00249-0
5. Audet, C., Le Digabel, S., Rochon Montplaisir, V., Tribes, C.: Algorithm 1027: NOMAD version 4: nonlinear optimization with the MADS algorithm. ACM Transactions on Mathematical Software **48**(3), 35:1–35:22 (2022). doi:10.1145/3544489
6. Cesaroni, E., Liuzzi, G., Lucidi, S.: Derivative-free bilevel optimization with inexact lower-level solutions. arXiv preprint arXiv:2603.20759 (2026). doi:10.48550/arXiv.2603.20759. URL `https://arxiv.org/abs/2603.20759`
7. Chen, L., Xu, J., Zhang, J.: On finding small hyper-gradients in bilevel optimization: hardness results and improved analysis. In: Proceedings of the 37th Conference on Learning Theory, *Proceedings of Machine Learning Research*, vol. 247, pp. 947–980. PMLR (2024). URL `https://proceedings.mlr.press/v247/chen24a.html`
8. Clarke, F.H.: Optimization and Nonsmooth Analysis, *Classics in Applied Mathematics*, vol. 5. Society for Industrial and Applied Mathematics (1990). doi:10.1137/1.9781611971309
9. Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. Annals of Operations Research **153**(1), 235–256 (2007). doi:10.1007/s10479-007-0176-2
10. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization, *MOS-SIAM Series on Optimization*, vol. 8. Society for Industrial and Applied Mathematics (2009). doi:10.1137/1.9780898718768
11. Conn, A.R., Vicente, L.N.: Bilevel derivative-free optimization and its application to robust optimization. Optimization Methods and Software **27**(3), 561–577 (2012). doi:10.1080/10556788.2010.547579
12. Cristofari, A., Rinaldi, F.: A derivative-free method for structured optimization problems. SIAM Journal on Optimization **31**(2), 1079–1107 (2021). doi:10.1137/20M1337417
13. Dempe, S.: Foundations of Bilevel Programming, *Nonconvex Optimization and Its Applications*, vol. 61. Springer (2002). doi:10.1007/b101970
14. Diouane, Y., Kungurtsev, V., Rinaldi, F., Zeffiro, D.: Inexact direct-search methods for bilevel optimization problems. Computational Optimization and Applications **88**(2), 469–490 (2024). doi:10.1007/s10589-024-00567-7

15. Dzahini, K.J., Rinaldi, F., Royer, C.W., Zeffiro, D.: Direct-search methods in the year 2025: theoretical guarantees and algorithmic paradigms. EURO Journal on Computational Optimization **13**, 100110 (2025). doi:10.1016/j.ejco.2025.100110
16. Ehrhardt, M.J., Roberts, L.: Inexact derivative-free optimization for bilevel learning. Journal of Mathematical Imaging and Vision **63**(5), 580–600 (2021). doi:10.1007/s10851-021-01020-8
17. Franceschi, L., Donini, M., Frasconi, P., Pontil, M.: Forward and reverse gradient-based hyperparameter optimization. In: Proceedings of the 34th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 70, pp. 1165–1173. PMLR (2017). URL `https://proceedings.mlr.press/v70/franceschi17a.html`
18. Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., Pontil, M.: Bilevel programming for hyperparameter optimization and meta-learning. In: Proceedings of the 35th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 80, pp. 1568–1577. PMLR (2018). URL `https://proceedings.mlr.press/v80/franceschi18a.html`
19. Goldstein, A.A.: Optimization of Lipschitz continuous functions. Mathematical Programming **13**(1), 14–22 (1977). doi:10.1007/BF01584320
20. Gratton, S., Royer, C.W., Vicente, L.N., Zhang, Z.: Direct search based on probabilistic feasible descent for bound and linearly constrained problems. Computational Optimization and Applications **72**(3), 525–559 (2019). doi:10.1007/s10589-019-00062-4
21. Kraft, D.: A software package for sequential quadratic programming. Technical report DFVLR-FB 88-28, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (1988)
22. Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numerica **28**, 287–404 (2019). doi:10.1017/S0962492919000060
23. Liu, Z., Chen, C., Luo, L., Low, B.K.H.: Zeroth-order methods for constrained nonconvex nonsmooth stochastic optimization. In: Proceedings of the 41st International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 235, pp. 30842–30872. PMLR (2024). URL `https://proceedings.mlr.press/v235/liu24j.html`
24. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. Computational Optimization and Applications **21**(2), 119–142 (2002). doi:10.1023/A:1013735414984
25. Maheshwari, C., Sastry, S.S., Ratliff, L., Mazumdar, E.: Convergent first-order methods for bi-level optimization and Stackelberg games. arXiv preprint arXiv:2302.01421v1 (2023). doi:10.48550/arXiv.2302.01421. URL `https://authors.library.caltech.edu/records/445x7-d7434`. Version 1
26. Menickelly, M., Wild, S.M.: Derivative-free robust optimization by outer approximations. Mathematical Programming **179**(1–2), 157–193 (2020). doi:10.1007/s10107-018-1326-9
27. Mersha, A.G., Dempe, S.: Direct search algorithm for bilevel programming problems. Computational Optimization and Applications **49**(1), 1–15 (2011). doi:10.1007/s10589-009-9295-9
28. Powell, M.J.D.: On fast trust region methods for quadratic models with linear constraints. Mathematical Programming Computation **7**(3), 237–267 (2015). doi:10.1007/s12532-015-0084-4
29. Ragonneau, T.M., Zhang, Z.: PDFO: a cross-platform package for Powell's derivative-free optimization solvers. Mathematical Programming Computation **16**(4), 535–559 (2024). doi:10.1007/s12532-024-00257-9
30. Rinaldi, F., Vicente, L.N., Zeffiro, D.: Stochastic trust-region and direct-search methods: a weak tail bound condition and reduced sample sizing. SIAM Journal on Optimization **34**(2), 2067–2092 (2024). doi:10.1137/22M1543446
31. Sahu, A.K., Zaheer, M., Kar, S.: Towards gradient free and projection free stochastic optimization. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, *Proceedings of Machine Learning Research*, vol. 89, pp. 3468–3477. PMLR (2019). URL `https://proceedings.mlr.press/v89/sahu19a.html`
32. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold,

J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods **17**(3), 261–272 (2020). doi:10.1038/s41592-019-0686-2

33. Ward, S., Zemkoho, A., Fisher, J., Benfield, D., Qian, Y.: BOLIB: bilevel optimisation library for benchmarking. Technical report, University of Southampton (2025). URL `https://www.bolib.org/about/`
34. Zhang, D., Lin, G.H.: Bilevel direct search method for leader–follower problems and application in health insurance. Computers & Operations Research **41**, 359–373 (2014). doi:10.1016/j.cor.2012.12.005
35. Zhang, Y., Khanduri, P., Tsaknakis, I., Yao, Y., Hong, M., Liu, S.: An introduction to bilevel optimization: foundations and applications in signal processing and machine learning. IEEE Signal Processing Magazine **41**(1), 38–59 (2024). doi:10.1109/MSP.2024.3358284
36. Zhou, S., Zemkoho, A.B., Tin, A.: BOLIB: bilevel optimization LIBrary of test problems. In: S. Dempe, A.B. Zemkoho (eds.) Bilevel Optimization: Advances and Next Challenges, *Springer Optimization and Its Applications*, vol. 161, pp. 563–580. Springer (2020). doi:10.1007/978-3-030-52119-6_19