

Bilevel optimization with a multi-objective lower-level problem: Risk-neutral and risk-averse formulations

T. Giovannelli* G. D. Kent† L. N. Vicente‡

February 11, 2024

Abstract

In this work, we propose different formulations and gradient-based algorithms for deterministic and stochastic bilevel problems with conflicting objectives in the lower level. Such problems have received little attention in the deterministic case and have never been studied from a stochastic approximation viewpoint despite the recent advances in stochastic methods for single-level, bilevel, and multi-objective optimization.

To solve bilevel problems with a multi-objective lower level, different approaches can be considered depending on the interpretation of the lower-level optimality. An optimistic formulation that was previously introduced for the deterministic case consists of minimizing the upper-level function over all non-dominated lower-level solutions. In this paper, we develop new risk-neutral and risk-averse formulations, address their main computational challenges, and develop the corresponding deterministic and stochastic gradient-based algorithms.

1 Introduction

In bilevel multi-objective optimization (BMO), at least one of the two levels of the bilevel problem has multiple objectives and can be modeled using the formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \quad & F_u(x, y) \\ \text{s.t.} \quad & x \in X \\ & y \in \operatorname{argmin}_{y \in Y(x)} F_\ell(x, y). \end{aligned} \tag{1.1}$$

The upper-level (UL) and lower-level (LL) objective functions $F_u : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ and $F_\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$ are given by $F_u = (f_u^1, \dots, f_u^p)$ and $F_\ell = (f_\ell^1, \dots, f_\ell^q)$, with f_u^i and f_ℓ^j real-valued functions for all $i \in \{1, \dots, p\}$ and $j \in \{1, \dots, q\}$, respectively, and $p \geq 1$ or $q \geq 1$. In this general formulation, the UL variables x are subjected to UL constraints ($x \in X$) and the LL variables y are subjected to LL constraints ($y \in Y(x)$); however, for the rest of this paper,

*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (tog220@lehigh.edu).

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (gdk220@lehigh.edu).

‡Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (lnv@lehigh.edu).

it is assumed that the LL variables are unconstrained (i.e., $Y(x) = \mathbb{R}^m$). The set X will be assumed closed and convex, which allows us to obtain feasible points with respect to the UL constraints by taking orthogonal projections onto X . In general, when at least one of the two levels is a multi-objective optimization problem, we can use the name bilevel multi-objective optimization [13, 14, 37]. When a multi-objective problem only arises in one of the two levels, one also finds the terminology semivectorial bilevel optimization [9].

Bilevel multi-objective optimization problems arise in applications related to defense, renewable energy systems, and fair machine learning. In the defense sector, a bilevel multi-objective problem for facility location was proposed in [25] to prevent an adversary from entering a territory by relocating wireless sensors in order to maximize the exposure of the attacker to the sensors (single-objective UL problem) and minimize the conflicting objectives given by the sensor relocation time and the total number of sensors (multi-objective LL problem). In [28], the authors propose a bilevel formulation to minimize the environmental impact of renewable energy systems and the resulting government expenditure (multi-objective UL problem) and the energy cost paid by the end users (single-objective LL problem). Bilevel optimization has recently been adopted to solve fair machine learning (ML) problems [24, 33, 34], where the goal is to minimize the prediction error on a validation dataset by training an ML model to avoid discriminatory predictions against people with sensitive attributes. To ensure accurate and fair prediction outcomes in real-life decision-making applications, accuracy and fairness loss functions must be jointly considered, thus leading to bilevel problems where both UL and LL problems are multi-objective.

In this paper, we consider bilevel multi-objective optimization problems with a multi-objective lower level (BMOLL), which can be obtained from problem (1.1) by considering $p = 1$ and $q > 1$. Hence, the problem to solve is

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \quad & f_u(x, y) \\ \text{s.t.} \quad & x \in X \\ & y \in \underset{y \in \mathbb{R}^m}{\operatorname{argmin}} F_\ell(x, y), \end{aligned} \tag{1.2}$$

where the UL objective function $f_u : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is real-valued. Throughout the paper, we assume f_u to be continuously differentiable and all the functions f_ℓ^j , with $j \in \{1, \dots, q\}$, to be twice continuously differentiable. We will formalize these assumptions in Section 4. Note that denoting $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ as the set-valued mapping representing the optimal solutions of the LL problem (we will identify this as the set of weak LL Pareto minimizers in Section 3), one can reformulate (1.2) as follows:

$$\underset{x \in \mathbb{R}^n, y \in \mathbb{R}^m}{\text{“min”}} \{f_u(x, y) \mid x \in X, y \in \Psi(x)\}, \tag{1.3}$$

where the notation “min” with double quotes is used in the literature to denote the ambiguity arising when there are multiple LL optimal solutions [9, 38]. Since the LL problem has conflicting objectives, given $x \in X$, a single optimal solution must be determined among the set $\Psi(x)$, and several criteria can be considered that potentially lead to different solutions of the bilevel problem. In the deterministic case, researchers have focused on *optimistic* formulations, excluding all those cases where the solution determined among the set of LL Pareto optimal points is not the most favorable for the UL problem.

As opposed to bilevel problems with UL and LL single-objective functions [4, 6, 7, 18, 19, 21, 39] (see also [3, 19, 26] for recent reviews), gradient-based methods for bilevel multi-objective problems have received less attention in the literature. Deterministic approaches for bilevel optimization with multi-objective LL problems have been proposed in [1, 9, 29, 30], where the LL problem is transformed into a single-objective problem by weighting the LL objective functions according to the weighted-sum approach that is utilized in multi-objective optimization [12]. The weights are then included among the UL optimization variables. Problems with multi-objective UL and LL problems ($p, q > 1$) have been addressed in [36], which uses the ε -constraint method [12] at both levels to obtain a single-objective bilevel problem. To the best of our knowledge, the only stochastic gradient-based algorithm for bilevel multi-objective problems has been proposed in [20], where multiple objectives are considered at the upper level. We point out that the problem solved in [20] is significantly different from problem (1.1) since in [20] there are multiple LL problems and each UL objective function only depends on the optimal solution of one LL problem. Moreover, in [20], the authors do not attempt to determine the UL Pareto front and consider a robust formulation of the UL problem to minimize the maximum optimal value among all the UL objective functions.

As an alternative to the known optimistic formulation developed for (1.2), we propose new *risk-neutral* and *risk-averse* formulations, address their computational challenges, and develop their corresponding deterministic and stochastic gradient descent algorithms. Both formulations are inspired by looking at y as a parameter, rather than as a variable. In the risk-neutral case, we minimize a new function describing the mean of the UL function over an x -dependent set of optimal LL solutions. We propose a formulation that is tractable (by rather taking the mean over LL weights) and can lead to efficient algorithms (by sampling the weights). The risk-averse case requires using the extension of Danskin’s Theorem to the case where the maximum is taken over an x -dependent efficient solution set.

This paper is organized as follows. We first review the bilevel stochastic gradient method in Section 2 (for a single objective in both the UL and LL). In Section 3, we introduce basic definitions, results, and general assumptions. We describe the known optimistic formulation for (1.2) in Section 4. The new risk-neutral and risk-averse formulations for (1.2), as well as the corresponding gradient-based algorithms, are introduced in Sections 5 and 6, respectively. Numerical results for synthetic bilevel problems with a multi-objective lower level are reported in Section 7, which also describes the practical implementations of the proposed methods. Finally, we draw some concluding remarks in Section 8, in particular how to develop the cases $p > 1, q = 1$ and $p, q > 1$ from known building blocks.

2 A review of bilevel stochastic gradient methods

Bilevel optimization with multi-objective upper or lower levels is based on (or uses as a reference) the bilevel single-objective (BO) case ($p = q = 1$), which can be modeled using the formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \quad & f_u(x, y) \\ \text{s.t.} \quad & x \in X \\ & y \in \operatorname{argmin}_{y \in \mathbb{R}^m} f_\ell(x, y). \end{aligned}$$

In BO problems, the goal of the UL problem is to minimize the UL objective function $f_u : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ over the UL variables x , which are subjected to UL constraints ($x \in X$), and LL variables y , which are subjected to being an optimal solution of the LL problem. The goal of the LL problem is to minimize the LL objective function $f_\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ over the LL variables y .

Assuming that there exists a unique solution $y(x)$ to the LL problem for all $x \in X$, problem BO is equivalent to a problem posed solely in terms of the UL variables x , and is given by

$$\min_{x \in \mathbb{R}^n} f(x) = f_u(x, y(x)) \quad \text{s.t.} \quad x \in X.$$

Recalling the assumptions that f_u is continuously differentiable, f_ℓ twice continuously differentiable, and further assuming $\nabla_{yy}^2 f_\ell(x, y(x))$ to be non-singular, the gradient of f at x can be obtained from the well-known adjoint (or hypergradient) formula

$$\nabla f = \nabla_x f_u - \nabla_{xy}^2 f_\ell (\nabla_{yy}^2 f_\ell)^{-1} \nabla_y f_u, \quad (2.1)$$

where all gradients and Hessians on the right-hand side are evaluated at $(x, y(x))$. The steepest descent direction for f at x is denoted by $d(x, y(x)) = -\nabla f(x)$. To obtain the adjoint formula, one can apply the chain rule to $f_u(x, y(x))$, which leads to $\nabla f = \nabla_x f_u + \nabla_y \nabla_y f_u$. Then, the Jacobian $\nabla y(x) \in \mathbb{R}^{n \times m}$ can be derived from the LL first-order necessary optimality conditions $\nabla_y f_\ell(x, y(x)) = 0$ by applying the chain rule to both sides of this equation with respect to x . The LL optimal solution function $y(\cdot)$ is continuously differentiable due to the implicit function theorem [35]. The equation resulting from the application of the chain rule is given by $\nabla_{yx}^2 f_\ell + \nabla_{yy}^2 f_\ell \nabla y^\top = 0$ (where all gradients and Hessians are evaluated at $(x, y(x))$), which leads to $\nabla y = -\nabla_{xy}^2 f_\ell (\nabla_{yy}^2 f_\ell)^{-1}$.

In Algorithm 1, we report a general framework for the bilevel stochastic gradient (BSG) method [19] for stochastic BO problems. Such a framework will be adapted to develop different algorithms for the BMOLL problem considered in this paper. We adopt ξ_k to denote the random variables used to obtain stochastic estimates for UL and LL gradients and Hessians. An initial point (x_0, y_0) and a sequence of positive scalars $\{\alpha_k\}$ are required as input. In Step 1, any appropriate optimization method can be applied to obtain an approximate LL solution \tilde{y} by solving the LL problem to a specified degree of accuracy. In Step 2, one computes an approximate negative BSG $d(x_k, \tilde{y}_k, \xi_k)$, defined by the adjoint gradient in (2.1), to update the UL variables. In Step 3, the vector x is updated by choosing a step size taken from the sequence of positive scalars $\{\alpha_k\}$. When X is a closed and convex constrained set different from \mathbb{R}^n , an orthogonal projection of $x_k + \alpha_k d(x_k, \tilde{y}_k, \xi_k)$ onto X is required (such a projection can be computed by solving a convex optimization problem). Regarding the stepsize sequence $\{\alpha_k\}$, popular options in the stochastic gradient literature are fixed or decaying stepsize sequences [2].

Algorithm 1 Bilevel Stochastic Gradient (BSG) Method

Input: $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^m$, $\{\alpha_k\}_{k \geq 0} > 0$.

For $k = 0, 1, 2, \dots$ **do**

Step 1. Obtain an approximation \tilde{y}_k to the LL optimal solution $y(x_k)$.

Step 2. Compute a negative BSG $d(x_k, \tilde{y}_k, \xi_k)$.

Step 3. Compute $x_{k+1} = P_X(x_k + \alpha_k d(x_k, \tilde{y}_k, \xi_k))$.

End do

As usual in the stochastic gradient literature [2], due to the lack of reasonable stopping criteria for stochastic algorithms, we do not include a stopping condition in Algorithm 1 (and in any of the algorithms developed in this paper). The convergence theory for the BSG method developed in [19] comprehensively covers several inexact settings, including the inexact solution of the LL problem and the use of noisy estimates of the gradients and Hessians involved. The convergence rates of the BSG method have been derived in [19] under the assumptions of non-convexity, strong convexity, and convexity of the true objective function f . The convergence theory of the algorithms introduced for the smooth case (i.e., optimistic and risk-neutral formulations in Sections 4-5, respectively) can be obtained as an extension of the convergence results presented in [19]. The development of the convergence theory for the non-smooth case (i.e., risk-averse formulation in Section 6) is left for future work.

3 Basic definitions, results, and general assumptions

Given $x \in X$, let us now focus on the multi-objective optimization problem given by the LL problem in (1.2). When the LL objective functions are conflicting, minimizing one objective results in worse values for at least one other objective. Therefore, there is typically no single optimal solution that minimizes all objective functions simultaneously. In such cases, one is interested in obtaining a set of points where the value of one objective cannot be improved without deteriorating the values of the other objectives. Points with this property are called strict/weak Pareto minimizers (or efficient or non-dominated points) and are introduced in Definitions 1-2 below, which adapt the standard definitions in [12] to the LL problem of (1.2).

Definition 1 (LL Pareto dominance) *Given any two points $\{y_1, y_2\} \subset \mathbb{R}^m$, we say that y_1 dominates y_2 if $F_\ell(x, y_1) < F_\ell(x, y_2)$ componentwise. Moreover, we say that y_1 weakly dominates y_2 if $F_\ell(x, y_1) \leq F_\ell(x, y_2)$ componentwise and $F_\ell(x, y_1) \neq F_\ell(x, y_2)$.*

Definition 2 (LL Pareto minimizer) *A point $y_* \in \mathbb{R}^m$ is a strict Pareto minimizer for the LL problem of (1.2) if no other point $\bar{y} \in \mathbb{R}^m$ exists such that y_* is weakly dominated by \bar{y} . A point $y_* \in \mathbb{R}^m$ is a weak Pareto minimizer if no other point $\bar{y} \in \mathbb{R}^m$ exists such that y_* is dominated by \bar{y} .*

Let $P_s(x)$ denote the set of strict LL Pareto minimizers and $P(x)$ the set of weak LL Pareto minimizers. We wish to highlight that $P(x)$ represents $\Psi(x)$ from (1.3). Mapping the set $P(x)$ into the objective space \mathbb{R}^q leads to the LL Pareto front, which is defined as $\{F_\ell(x, y) : y \in P(x)\}$. Note that Definition 2 implies that $P_s(x) \subseteq P(x)$.

To compute a Pareto front, one can use scalarization techniques to reduce a multi-objective problem into a single-objective one, which can then be solved using classical optimization approaches [12, 31]. One popular scalarization technique is the weighted-sum method, which consists of weighting the objective functions into a single objective $\sum_{j=1}^q \lambda_j f_\ell^j(x, y)$, where λ_j are non-negative weights, for all $j \in \{1, \dots, q\}$. A necessary and sufficient condition for weak LL Pareto optimality based on the weighted-sum method is included in Proposition 3.1 below, along with a sufficient condition for equivalence between $P_s(x)$ and $P(x)$. We refer to [12, 17, 31] for the proof of such a proposition.

Proposition 3.1 *Let the LL objective functions $f_\ell^1(x, \cdot), \dots, f_\ell^q(x, \cdot)$ be convex for all $x \in X$. Then, $y_* \in P(x)$ if and only if there exist weights $\lambda_j \geq 0$, for all $j \in \{1, \dots, q\}$, not all zero, such that $y_* \in \operatorname{argmin}_{y \in \mathbb{R}^m} \sum_{j=1}^q \lambda_j f_\ell^j(x, y)$. Moreover, if the LL objective functions $f_\ell^1(x, \cdot), \dots, f_\ell^q(x, \cdot)$ are strictly convex for a certain $x \in X$, then $P_s(x) = P(x)$.*

For the remainder of the paper, we require Assumption 3.1 below.

Assumption 3.1 *The LL objective functions $f_\ell^1(x, \cdot), \dots, f_\ell^q(x, \cdot)$ are strictly convex for all $x \in X$. Further, the set $P(x)$ is non-empty for all $x \in X$.*

Under Assumption 3.1, it is clear from Proposition 3.1 that $P(x) = P_s(x)$. The following remark characterizes the non-emptiness of $P(x)$.

Remark 3.1 *In the case $Y(x) = \mathbb{R}^m$ considered in this paper, we point out that $P(x) \neq \emptyset$ if the LL objective functions $f_\ell^1(x, \cdot), \dots, f_\ell^q(x, \cdot)$ are uniformly convex, which further implies that $P(x)$ is also compact. When $Y(x) \neq \mathbb{R}^m$, we have that $P_s(x) \neq \emptyset$ (and thus $P(x) \neq \emptyset$) if the set $Y(x)$ is compact and non-empty, in which case $P(x)$ is also compact.*

4 The known optimistic formulation

The so-called optimistic formulation of problem (1.2) corresponds to the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \quad & f_u(x, y) \\ \text{s.t.} \quad & x \in X \\ & y \in P(x), \end{aligned} \tag{4.1}$$

where the set of LL optimal solutions is given by the set of weak LL Pareto minimizers $P(x)$. In accordance with Proposition 4.1 below, one can prove that problem (4.1) is equivalent to

$$\begin{aligned} \min_{x \in \mathbb{R}^n, \lambda \in \mathbb{R}^q, y \in \mathbb{R}^m} \quad & f_u(x, y) \\ \text{s.t.} \quad & x \in X, \lambda \in \Lambda \\ & y \in \operatorname{argmin}_{y \in \mathbb{R}^m} f_\ell(x, \lambda, y) := \lambda^\top F_\ell(x, y), \end{aligned} \tag{4.2}$$

where Λ denotes the simplex set, i.e.,

$$\Lambda = \left\{ \lambda \in \mathbb{R}^q \mid \sum_{j=1}^q \lambda_j = 1, \lambda_j \geq 0 \forall j \in \{1, \dots, q\} \right\}. \tag{4.3}$$

Given $(x, \lambda) \in X \times \Lambda$, let $\Phi(x, \lambda) = \{y \in \mathbb{R}^m \mid y \in \operatorname{argmin}_{y \in \mathbb{R}^m} f_\ell(x, \lambda, y)\}$ denote the set of optimal solutions to the LL problem in (4.2). The equivalence between problems (4.1) and (4.2) is stated below (assuming implicitly that each problem admits an optimal solution).

Proposition 4.1 *Let Assumption 3.1 hold. If (\bar{x}, \bar{y}) is an optimal solution to problem (4.1), then, for all $\bar{\lambda} \in \Lambda$ such that $\bar{y} \in \Phi(\bar{x}, \bar{\lambda})$, the point $(\bar{x}, \bar{\lambda}, \bar{y})$ is an optimal solution to problem (4.2). Moreover, if $(\bar{x}, \bar{\lambda}, \bar{y})$ is an optimal solution to problem (4.2), then (\bar{x}, \bar{y}) is an optimal solution to problem (4.1).*

Proof. Let (\bar{x}, \bar{y}) be an optimal solution to problem (4.1). We have $(\bar{x}, \bar{y}) \in X \times P(\bar{x})$ and $f_u(\bar{x}, \bar{y}) \leq f_u(x, y)$ for all $(x, y) \in X \times P(x)$. Assume that there exists $\lambda^0 \in \Lambda$ with $\bar{y} \in \Phi(\bar{x}, \lambda^0)$ such that $(\bar{x}, \lambda^0, \bar{y})$ is not an optimal solution to problem (4.2). Therefore, there exists $(\hat{x}, \hat{\lambda}) \in X \times \Lambda$, with $\hat{y} \in \Phi(\hat{x}, \hat{\lambda})$, such that $f_u(\bar{x}, \bar{y}) > f_u(\hat{x}, \hat{y})$. Note that from Proposition 3.1, since $\hat{y} \in \Phi(\hat{x}, \hat{\lambda})$, it follows that $\hat{y} \in P(\hat{x})$, which implies that (\hat{x}, \hat{y}) is a feasible point for problem (4.1). All these facts contradict the optimality of (\bar{x}, \bar{y}) for problem (4.1).

Vice versa, let $(\bar{x}, \bar{\lambda}, \bar{y})$ be an optimal solution to problem (4.2). We have $(\bar{x}, \bar{\lambda}) \in X \times \Lambda$, with $\bar{y} \in \Phi(\bar{x}, \bar{\lambda})$, and $f_u(\bar{x}, \bar{y}) \leq f_u(x, y)$ for all $(x, \lambda) \in X \times \Lambda$, with $y \in \Phi(x, \lambda)$. From Proposition 3.1, since $\bar{y} \in \Phi(\bar{x}, \bar{\lambda})$, it follows that $\bar{y} \in P(\bar{x})$, which implies that (\bar{x}, \bar{y}) is a feasible point for problem (4.1). Assume that (\bar{x}, \bar{y}) is not an optimal solution to problem (4.1). Therefore, there exists $(\hat{x}, \hat{y}) \in X \times P(\hat{x})$ such that $f_u(\bar{x}, \bar{y}) > f_u(\hat{x}, \hat{y})$. From Proposition 3.1, since $\hat{y} \in P(\hat{x})$, there exists $\hat{\lambda} \in \Lambda$ such that $\hat{y} \in \Phi(\hat{x}, \hat{\lambda})$, which implies that $(\hat{x}, \hat{\lambda}, \hat{y})$ is a feasible point for problem (4.2). All these facts contradict the optimality of $(\bar{x}, \bar{\lambda}, \bar{y})$ for problem (4.2). \square

Note that the LL objective function in problem (4.2) is $f_\ell(x, \lambda, y) = \sum_{j=1}^q \lambda_j f_\ell^j(x, y)$, where $\lambda \in \Lambda$. Hence, problem (4.2) is a bilevel problem where both levels are single-objective functions and can be solved by applying the BSG method introduced in Section 2 (when the functions are stochastic). Let us denote the optimal solution of the LL problem in (4.2) by $y(x, \lambda)$. For the calculation of the BSG direction, we require Assumption 4.2 below, which ensures the existence of $y(x, \lambda)$ for all $\lambda \in \Lambda$. To do this, we formally assume a certain smoothness of the functions we are dealing with.

Assumption 4.1 *The UL function f_u is once continuously differentiable and all the LL functions f_ℓ^j , with $j \in \{1, \dots, q\}$, are twice continuously differentiable.*

We remark that Assumptions 3.1 and 4.1 together imply that the Hessians $\nabla_{yy}^2 f_\ell^j(x, y)$, for all $j \in \{1, \dots, q\}$, are positive definite for all $x \in X$.

Further, we also assume the existence of a solution to the LL problem. We state here this requirement for both the optimistic and risk-neutral formulations together (although we formally introduce the risk-neutral case in Section 5) to avoid repetition. It bears mentioning that this assumption does not encompass the risk-averse case here, as it requires a different approach, which we introduce in Section 6.

Assumption 4.2 (Existence of LL solution) *For any $x \in X$ and $\lambda \in \Lambda$, there exists a point $y(x, \lambda)$ such that $\nabla_y f_\ell(x, \lambda, y(x, \lambda)) = \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y(x, \lambda)) = 0$. Moreover, the stochastic estimates of the Hessians $\nabla_{yy}^2 f_\ell^j(x, y)$, for all $j \in \{1, \dots, q\}$, are positive definite at all points.*

Given the strict convexity of the LL functions imposed in Assumption 3.1, it then becomes clear under Assumption 4.2 that the point $y(x, \lambda)$ is the unique solution of the LL problem.

Let $f_{\text{OPT}}(x, \lambda) = f_u(x, y(x, \lambda))$, with OPT standing for optimistic. By applying the chain rule to $f_u(x, y(x, \lambda))$, one obtains the gradient vectors

$$\nabla_x f_{\text{OPT}} = \nabla_x f_u + \nabla_{xy} \nabla_y f_u, \quad \nabla_\lambda f_{\text{OPT}} = \nabla_{\lambda y} \nabla_y f_u. \quad (4.4)$$

To calculate the Jacobian $\nabla y(x, \lambda) \in \mathbb{R}^{(n+q) \times m}$, we take derivatives with respect to x and λ on both sides of the LL first-order necessary optimality conditions $\nabla_y f_\ell(x, \lambda, y(x, \lambda)) = \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y(x, \lambda)) = 0$, yielding the equations

$$\nabla_{xx}^2 f_\ell + \nabla_{yy}^2 f_\ell \nabla_x y^\top = 0, \quad \nabla_{y\lambda}^2 f_\ell + \nabla_{yy}^2 f_\ell \nabla_\lambda y^\top = 0. \quad (4.5)$$

Again, the differentiability of $y(\cdot)$ with respect to x and λ is a consequence of the implicit function theorem [35]. Under Assumption 4.2, we can obtain $\nabla_x y$ and $\nabla_\lambda y$ from (4.5) and plug their values into (4.4), which leads to

$$\nabla_x f_{\text{OPT}} = \nabla_x f_u - \nabla_{xy}^2 f_\ell (\nabla_{yy}^2 f_\ell)^{-1} \nabla_y f_u, \quad \nabla_\lambda f_{\text{OPT}} = -\nabla_{\lambda y}^2 f_\ell (\nabla_{yy}^2 f_\ell)^{-1} \nabla_y f_u, \quad (4.6)$$

where all gradients and Hessians are evaluated at $(x, y(x, \lambda))$. Note that

$$\nabla_{xy}^2 f_\ell = \sum_{j=1}^q \lambda_j \nabla_{xy}^2 f_\ell^j, \quad \nabla_{yy}^2 f_\ell = \sum_{j=1}^q \lambda_j \nabla_{yy}^2 f_\ell^j, \quad \nabla_{\lambda y}^2 f_\ell = (\nabla_y f_\ell^1, \dots, \nabla_y f_\ell^q)^\top. \quad (4.7)$$

Therefore, from (4.6)–(4.7), one can obtain the adjoint gradient ∇f_{OPT} by concatenating the subvectors $\nabla_x f_{\text{OPT}}$ and $\nabla_\lambda f_{\text{OPT}}$ into a single vector. In the stochastic case, all the gradients and Hessians on the right-hand sides of (4.6) can be replaced by corresponding stochastic estimates.

In Algorithm 2, we introduce a bilevel stochastic gradient method to solve the optimistic formulation of problem (1.2), given by problem (4.2). Note that the main differences between Algorithm 2 and the classical BSG method reported in Algorithm 1 are Step 2, where one now computes a (negative) BSG $d(x_k, \lambda_k, \tilde{y}_k, \xi_k)$ to approximate $-\nabla f_{\text{OPT}} = -(\nabla_x f_{\text{OPT}}, \nabla_\lambda f_{\text{OPT}})$, and Step 3, where the orthogonal projection is now applied to the λ variables as well.

Algorithm 2 BSG-OPT Method

Input: $(x_0, \lambda_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^m$, $\{\alpha_k\}_{k \geq 0} > 0$.

For $k = 0, 1, 2, \dots$ **do**

Step 1. Obtain an approximation \tilde{y}_k to the LL optimal solution $y(x_k, \lambda_k)$.

Step 2. Compute a negative BSG $d(x_k, \lambda_k, \tilde{y}_k, \xi_k)$.

Step 3. Compute $(x_{k+1}, \lambda_{k+1}) = P_{X\Lambda}((x_k, \lambda_k) + \alpha_k d(x_k, \lambda_k, \tilde{y}_k, \xi_k))$, where $P_{X\Lambda}$ projects the x and λ variables onto the feasible regions X and Λ , respectively.

End do

Note that, in principle, every point $y \in P(x)$ can be considered an LL optimal solution. Different LL Pareto points have a different impact on the UL objective function and, accordingly, they can lead to different optimal solutions to the bilevel problem. However, by using the optimistic formulation (4.2), only the LL Pareto point that is most favorable for the UL objective function is selected among all the points $y \in P(x)$. Thus, we will now consider new alternative approaches to the optimistic formulation (4.2).

5 A new risk-neutral formulation

In this section, we introduce a new formulation for problem (1.2). To gain intuition, suppose that the set of weak LL Pareto minimizers is the same for all feasible values of x , i.e., $P(x) = P$ for all $x \in X$. By interpreting y as a parameter, one can consider the parametric optimization problem

$$\left. \begin{array}{l} \min_{x \in \mathbb{R}^n} f_u(x, y) \\ \text{s.t.} \quad x \in X \end{array} \right\} \text{ with } y \in P, \quad (5.1)$$

which can be addressed by considering two approaches in addition to the optimistic one. The *risk-neutral* approach assumes that y is a random vector with a probability distribution defined over P and considers the formulation

$$\min_{x \in X} \mathbb{E}_{y \sim P} [f_u(x, y)], \quad (5.2)$$

where $\mathbb{E}_{y \sim P}$ denotes the expected value that is taken with respect to the distribution of y over the domain P . The objective function of problem (5.2) can be approximated by using a sample mean, and the resulting problem can be solved by applying the SG method.

In the risk-neutral formulation for the general case with $P(x)$, given x and assuming that y is a random vector with a probability distribution defined over $P(x)$, the problem to solve is

$$\min_{x \in X} \mathbb{E}_{y \sim P(x)} [f_u(x, y)]. \quad (5.3)$$

Under Assumption 3.1, we consider a companion problem to (5.3) that will provide us with a tractable solution procedure. In particular, assuming that λ is a continuous random vector with a probability distribution defined over Λ (defined in (4.3)), the problem we will consider instead is

$$\min_{x \in X} \mathbb{E}_{\lambda \sim \Lambda} [f_u(x, y(x, \lambda))], \quad (5.4)$$

where $y(x, \lambda)$ now denotes the optimal solution of the problem

$$\min_{y \in \mathbb{R}^m} F_\ell(x, y)^\top \lambda. \quad (5.5)$$

Again, Assumptions 3.1, 4.1, and 4.2 all together imply that (5.5) has a unique solution.

Under Assumptions 3.1, 4.1, and 4.2, one can also calculate the BSG direction for the risk-neutral formulation. In practice, one can consider a finite set $\Lambda_{\text{RN}} = \{\lambda^1, \dots, \lambda^N\} \subset \Lambda$ that corresponds to a fine-scale discretization of Λ , with RN standing for risk-neutral. The corresponding set of weak LL Pareto minimizers is $\{y(x, \lambda^1), \dots, y(x, \lambda^N)\} \subset P(x)$. Therefore, problem (5.4) can be approximated as

$$\min_{x \in X} f_{\text{RN}}(x) = \frac{1}{N} \sum_{i=1}^N f_u(x, y(x, \lambda^i)). \quad (5.6)$$

Note that the gradient of the objective function in (5.6) is given by

$$\nabla f_{\text{RN}}(x) = \frac{1}{N} \sum_{i=1}^N \nabla f_{\text{RN}}^i(x), \quad (5.7)$$

where $f_{\text{RN}}^i(x) = f_u(x, y(x, \lambda^i))$ for all $i \in \{1, \dots, N\}$.

By applying the chain rule to $f_u(x, y(x, \lambda^i))$, one obtains

$$\nabla f_{\text{RN}}^i = \nabla_x f_u + \nabla_x y(x, \lambda^i) \nabla_y f_u. \quad (5.8)$$

Then, the Jacobian $\nabla_x y(x, \lambda^i) \in \mathbb{R}^{n \times m}$ can be calculated through the first-order LL Pareto necessary optimality condition $\sum_{j=1}^q \lambda_j^i \nabla_y f_\ell^j(x, y(x, \lambda^i)) = 0$, where λ_j^i denotes the j -th component of the vector λ^i , for all $i \in \{1, \dots, N\}$. In particular, by taking the derivative of both sides of this equation with respect to x , using the chain rule and the implicit function theorem, we obtain

$$\sum_{j=1}^q \lambda_j^i \left(\nabla_{yx}^2 f_\ell^j + \nabla_{yy}^2 f_\ell^j \nabla_x y(x, \lambda^i)^\top \right) = 0, \quad (5.9)$$

where all Hessians are evaluated at $(x, y(x, \lambda^i))$. We recall here that Assumptions 3.1 and 4.1 together imply that any convex combination of the Hessians $\nabla_{yy}^2 f_\ell^j(x, y)$, $j \in \{1, \dots, q\}$, is positive definite, and thus non-singular. Equation (5.9) yields

$$\nabla_x y(x, \lambda^i) = - \left(\sum_{j=1}^q \lambda_j^i \nabla_{xy}^2 f_\ell^j \right) \left(\sum_{j=1}^q \lambda_j^i \nabla_{yy}^2 f_\ell^j \right)^{-1}, \quad \text{for all } i \in \{1, \dots, N\}. \quad (5.10)$$

Plugging (5.10) into (5.8), one obtains

$$\nabla f_{\text{RN}}^i = \nabla_x f_u - \left(\sum_{j=1}^q \lambda_j^i \nabla_{xy}^2 f_\ell^j \right) \left(\sum_{j=1}^q \lambda_j^i \nabla_{yy}^2 f_\ell^j \right)^{-1} \nabla_y f_u, \quad (5.11)$$

where all gradients and Hessians on the right-hand side are evaluated at $(x, y(x, \lambda^i))$. In the stochastic case, all the gradients and Hessians on the right-hand side of (5.11) can be replaced by corresponding stochastic estimates.

Since the number of elements in Λ_{RN} can be significantly large, one can apply SG to solve problem (5.6) by randomly choosing a set of samples (i.e., a mini-batch) from Λ_{RN} . Denoting a mini-batch as $\Lambda_Q = \{\lambda^1, \dots, \lambda^Q\} \subseteq \Lambda_{\text{RN}}$, where Q is the mini-batch size, the corresponding sample set of LL Pareto minimizers $\{y(x, \lambda^1), \dots, y(x, \lambda^Q)\}$ can be used to compute a SG for ∇f_{RN} in (5.7).

In Algorithm 3, we introduce a bilevel stochastic gradient method to solve the risk-neutral interpretation of problem (1.2) given by the formulation (5.6). We adopt $(\xi_i)_k$ to denote the random variables used to obtain stochastic estimates for the UL and LL gradients and Hessians, for all $i \in \{1, \dots, Q\}$. For the sake of simplicity, at Steps 2–3, we denote $\tilde{y}_k = \{(\tilde{y}(x_k, \tilde{\lambda}^i) \mid i \in \{1, \dots, Q\})\}$ and $\xi_k = \{(\xi_i)_k \mid i \in \{1, \dots, Q\}\}$.

Algorithm 3 BSG-RN Method

Input: $x_0 \in \mathbb{R}^n$, $\{\alpha_k\}_{k \geq 0} > 0$.

For $k = 0, 1, 2, \dots$ **do**

Step 1. Obtain a mini-batch $\{\tilde{\lambda}^1, \dots, \tilde{\lambda}^Q\}$ and determine $\{\tilde{y}(x_k, \tilde{\lambda}^1), \dots, \tilde{y}(x_k, \tilde{\lambda}^Q)\}$ to approximate the LL Pareto optimal points $\{y(x_k, \tilde{\lambda}^1), \dots, y(x_k, \tilde{\lambda}^Q)\}$.

Step 2. Obtain a negative BSG $d(x_k, \tilde{y}(x_k, \tilde{\lambda}^i), (\xi_i)_k)$ to approximate $-\nabla f_{\text{RN}}^i(x_k)$ for all $i \in \{1, \dots, Q\}$ and compute $d(x_k, \tilde{y}_k, \xi_k) = (1/Q) \sum_{i=1}^Q d(x_k, \tilde{y}(x_k, \tilde{\lambda}^i), (\xi_i)_k)$.

Step 3. Compute $x_{k+1} = P_X(x_k + \alpha_k d(x_k, \tilde{y}_k, \xi_k))$.

End do

6 A new risk-averse formulation

In this section, we introduce another new formulation for problem (1.2). Again, to gain intuition, suppose $P(x) = P$ for all $x \in X$, and consider problem (5.1). The *risk-averse* (or robust or pessimistic) formulation is given by the problem

$$\min_{x \in X} \max_{y \in P} f_u(x, y),$$

which can be reformulated as $\min_{x \in X} f_{\text{RA}}(x)$ by introducing $f_{\text{RA}}(x) = \max_{y \in P} f_u(x, y)$, with RA standing for risk-averse. Since f_{RA} is nonsmooth, one can solve such a problem by applying a stochastic subgradient algorithm, where a subgradient can be obtained from the subdifferential of f_{RA} at x . Based on Danskin's Theorem [8], such a subdifferential is given by $\partial f_{\text{RA}}(x) = \text{conv}\{\nabla_x f_u(x, y) \mid y \in Y_0(x)\}$, where $Y_0(x) = \{\bar{y} \in P \mid f_u(x, \bar{y}) = \max_{y \in P} f_u(x, y)\}$.

In the risk-averse formulation for the general case with $P(x)$, the problem to solve is

$$\min_{x \in X} \max_{y \in P(x)} f_u(x, y), \quad (6.1)$$

which can be reformulated as $\min_{x \in X} f_{\text{RA}}(x)$ by introducing $f_{\text{RA}}(x) = \max_{y \in P(x)} f_u(x, y)$. Under Assumptions 3.1 and 4.1, let us specify $P(x)$ algebraically by using the LL first-order necessary and sufficient conditions for Pareto optimality (see Proposition 3.1) as follows:

$$P(x) = \left\{ y \in \mathbb{R}^m \mid \exists \lambda \in \Lambda \text{ such that } \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y) = 0 \right\}. \quad (6.2)$$

In accordance with Proposition 6.1 below (where we will implicitly assume the well definedness of each problem), one can prove that Problem (6.1) is equivalent to the problem

$$\min_{x \in X} \max_{(y, \lambda) \in \tilde{P}(x)} f_u(x, y), \quad (6.3)$$

where

$$\tilde{P}(x) = \left\{ (y, \lambda) \in \mathbb{R}^m \times \mathbb{R}^q \mid \lambda \in \Lambda \text{ and } \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y) = 0 \right\}. \quad (6.4)$$

Problem (6.3) can be reformulated as $\min_{x \in X} \tilde{f}_{\text{RA}}(x)$ by introducing the function $\tilde{f}_{\text{RA}}(x) = \max_{(y, \lambda) \in \tilde{P}(x)} f_u(x, y)$. The optimization problem defining $\tilde{f}_{\text{RA}}(x)$ can be written as

$$\begin{aligned} & \max_{y \in \mathbb{R}^m, \lambda \in \mathbb{R}^q} f_u(x, y) \\ & \text{s.t. } \lambda \in \Lambda, \\ & \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y) = 0. \end{aligned} \tag{6.5}$$

Note that from the definitions of $P(x)$ and $\tilde{P}(x)$ in (6.2) and (6.4), respectively, we have

$$y \in P(x) \text{ if and only if } (y, \lambda) \in \tilde{P}(x) \text{ for some } \lambda \in \Lambda. \tag{6.6}$$

Proposition 6.1 *\bar{x} is an optimal solution of problem (6.1) if and only if \bar{x} is an optimal solution of problem (6.3).*

Proof. Let \bar{x} be an optimal solution of problem (6.1). Therefore, we can write $f_{\text{RA}}(\bar{x}) \leq f_{\text{RA}}(\tilde{x})$ for all $\tilde{x} \in X$ or, equivalently, $f_u(\bar{x}, \bar{y}) \leq f_u(\tilde{x}, \tilde{y})$ for all $\tilde{x} \in X$, where $\bar{y} \in \operatorname{argmax}_{y \in P(\bar{x})} f_u(\bar{x}, y)$ and $\tilde{y} \in \operatorname{argmax}_{y \in P(\tilde{x})} f_u(\tilde{x}, y)$. Now, assume that \bar{x} is not a minimizer of the outer minimization problem in (6.3). Then, there exists a point $\hat{x} \in X$ such that $\tilde{f}_{\text{RA}}(\bar{x}) > \tilde{f}_{\text{RA}}(\hat{x})$ or, equivalently, $f_u(\bar{x}, \bar{y}) > f_u(\hat{x}, \hat{y})$, with $\hat{y} \in \operatorname{argmax}_{(y, \lambda) \in \tilde{P}(\hat{x})} f_u(\hat{x}, y)$. From (6.6) and from the fact that the objective function of the inner maximization problem in (6.3) does not depend on λ , it follows that $\hat{y} \in \operatorname{argmax}_{y \in P(\hat{x})} f_u(\hat{x}, y)$. Therefore, $f_{\text{RA}}(\bar{x}) > f_{\text{RA}}(\hat{x})$, which contradicts the optimality of \bar{x} for problem (6.1).

Now, let \bar{x} be an optimal solution of problem (6.3). Therefore, $\tilde{f}_{\text{RA}}(\bar{x}) \leq \tilde{f}_{\text{RA}}(\tilde{x})$ for all $\tilde{x} \in X$ or, equivalently, $f_u(\bar{x}, \bar{y}) \leq f_u(\tilde{x}, \tilde{y})$ for all $\tilde{x} \in X$, where $\bar{x} \in \operatorname{argmax}_{(y, \lambda) \in \tilde{P}(\bar{x})} f_u(\bar{x}, y)$ and $\tilde{y} \in \operatorname{argmax}_{(y, \lambda) \in \tilde{P}(\tilde{x})} f_u(\tilde{x}, y)$. Again, from (6.6) and from the fact that the objective function of the inner maximization problem in (6.3) does not depend on λ , we have $\tilde{y} \in \operatorname{argmax}_{y \in P(\tilde{x})} f_u(\tilde{x}, y)$. Therefore, $f_{\text{RA}}(\bar{x}) \leq f_{\text{RA}}(\tilde{x})$ for all $\tilde{x} \in X$, which shows that \bar{x} is an optimal solution for problem (6.1). \square

For the calculation of a BSG direction, and under Assumption 4.1, we start by introducing the Lagrangian function of problem (6.5) as $\mathcal{L}(x, y, \lambda, z) = f_u(x, y) + z_I^\top \lambda + z_E^\top (\sum_{i=1}^q \lambda_i - 1, \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x, y))$, where $z_I \in \mathbb{R}^q$ and $z_E \in \mathbb{R}^{m+1}$ are the vectors of Lagrange multipliers associated with the inequality and equality constraints, respectively. We will assume below the satisfaction of the first-order KKT conditions and the linear independence of the gradients of the active constraints (LICQ) [32] for problem (6.5).

Assumption 6.1 *For all $x \in X$, there exists a $(y(x), \lambda(x))$ satisfying the first-order KKT conditions for problem (6.5) with associated multipliers $(z_I(x), z_E(x))$ such that the LICQ is satisfied.*

Based on [15, Corollary 4.11], under Assumption 6.1 and additionally requiring $\tilde{P}(x)$ to be non-empty and compact, the subdifferential of \tilde{f}_{RA} at x is given by

$$\partial \tilde{f}_{\text{RA}}(x) = \operatorname{conv}\{\nabla_x \mathcal{L}(x, y(x), \lambda(x), z(x, y(x), \lambda(x))) \mid (y(x), \lambda(x)) \in \tilde{Y}_0(x)\}, \tag{6.7}$$

where $\tilde{Y}_0(x) = \{(\bar{y}(x), \bar{\lambda}(x)) \in \tilde{P}(x) \mid f_u(x, \bar{y}(x)) = \max_{(y, \lambda) \in \tilde{P}(x)} f_u(x, y)\}$ and $z(x, y(x), \lambda(x))$ is the unique optimal vector of Lagrange multipliers (the uniqueness is a trivial consequence of Assumption 6.1). One can ensure the compactness of $\tilde{P}(x)$ from (6.6) by noticing that Λ is compact and requiring $P(x)$ to be compact, which can be satisfied as suggested in Remark 3.1. The non-emptiness of $\tilde{P}(x)$ (assumed in Assumption 3.1 for $P(x)$) is ensured similarly; see also Remark 3.1. Note that the gradient of the Lagrangian with respect to x is given by

$$\nabla_x \mathcal{L}(x, y, \lambda, z) = \nabla_x f_u(x, y) + \sum_{j=1}^q \lambda_j \nabla_{xy}^2 f_\ell^j(x, y) z_E. \quad (6.8)$$

Equation (6.8) can be used in (6.7), where $\tilde{Y}_0(x)$ is given by the solutions $(\bar{y}(x), \bar{\lambda}(x))$ of problem (6.5). In the stochastic case, all the gradients and Hessians on the right-hand side of (6.7)–(6.8) can be replaced by corresponding stochastic estimates. When using stochastic estimates, the subdifferential (6.7) is denoted as $\partial \tilde{f}_{\text{RA}}^d(x)$.

In Algorithm 4, we introduce a bilevel stochastic subgradient method to solve the risk-averse formulation of problem (1.2) given by problem (6.1). We adopt ξ_k to denote the random variables used to obtain stochastic estimates for UL and LL gradients and Hessians.

Algorithm 4 BSG-RA Method

Input: $x_0 \in \mathbb{R}^n$, $\{\alpha_k\}_{k \geq 0} > 0$.

For $k = 0, 1, 2, \dots$ **do**

Step 1. Obtain an approximation $(\tilde{y}_k, \tilde{\lambda}_k)$ to a solution $(y(x_k), \lambda(x_k))$ of problem (6.5).

Step 2. Select a negative stochastic subgradient $d(x_k, \tilde{y}_k, \tilde{\lambda}_k, \xi_k) \in -\partial \tilde{f}_{\text{RA}}^d(x_k)$.

Step 3. Compute $x_{k+1} = P_X(x_k + \alpha_k d(x_k, \tilde{y}_k, \tilde{\lambda}_k, \xi_k))$.

End do

7 Numerical experiments

All code was written in Python and the experimental results were obtained on a desktop computer (32GB of RAM, Intel(R) Core(TM) i9-9900K processor running at 3.60GHz).*

7.1 Our practical methods

In the numerical experiments, we tested Algorithms 2–4 with both exact and stochastic Hessians in deterministic and stochastic settings, respectively. The resulting algorithms are referred to as BSG-OPT-H, BSG-RN-H, and BSG-RA-H, where the “H” stands for the Hessian matrix. To deal with the inverse matrix in (4.6), one could solve the linear system given by the adjoint equation $(\sum_{j=1}^q \lambda_j \nabla_{yy}^2 f_\ell^j) \mu = \nabla_y f_u$ for the variables μ , and then calculate $\nabla_x f_{\text{OPT}}$ from $\nabla_x f_u - (\nabla_{xy}^2 f_\ell) \mu$ and $\nabla_\lambda f_{\text{OPT}}$ from $-(\nabla_{\lambda y}^2 f_\ell) \mu$. A similar approach could be used to handle the inverse term in (5.11). Due to the small dimensions of the problems that we tested ($n, m \leq 50$), in BSG-OPT and BSG-RN, the adjoint systems are solved by factorizing the matrices $\sum_{j=1}^q \lambda_j \nabla_{yy}^2 f_\ell^j$ and $\sum_{j=1}^q \lambda_j \nabla_{yy}^2 f_\ell^j$, respectively. In practice, when the dimensions of the problems are large,

*All the code for our implementation is available at https://github.com/GdKent/BMOLL_OPT_RN_RA.

Problem	n	m	Ref. for LL	LL Objective Functions	Bound on x_i
1	\bar{n}	\bar{n}	JOS1 [23]	$f_\ell^1(x, y) = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} x_i^2 y_i^2$ $f_\ell^2(x, y) = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} (x_i - 2)^2 (y_i - 2)^2$	$[-2, \infty]$
2	\bar{n}	\bar{n}	SP1 [22]	$f_\ell^1(x, y) = \sum_{i=1}^{\bar{n}} [(x_i - 1)^2 + (x_i - y_i)^2]$ $f_\ell^2(x, y) = \sum_{i=1}^{\bar{n}} [(y_i - 3)^2 + (x_i - y_i)^2]$	$[-2, 3]$
3	\bar{n}	\bar{n}	GKV1	$f_\ell^1(x, y) = \frac{1}{2} y^\top H_3 y - \frac{1}{2} y^\top H_4 x$ $f_\ell^2(x, y) = \frac{1}{2} y^\top H_5 y + \frac{1}{2} y^\top H_6 x$	$[-\infty, 0]$ or $[0, \infty]$

Table 1: Test problems (\bar{n} is an arbitrary positive scalar).

one can solve the adjoint systems via the linear conjugate gradient method until non-positive curvature is detected. In Step 1 of BSG-OPT and BSG-RN, we apply either gradient descent (in the deterministic setting) or stochastic gradient descent (in the stochastic setting) for a certain budget of iterations. The number of iterations increases by 1 every time the difference of the UL objective function between two consecutive iterations is less than a given threshold. Such an increasing accuracy strategy has been used successfully in the BSG method presented in [19]. In Step 1 of BSG-RA, we solve problem (6.5) by applying the trust-region algorithm for nonlinear constrained problems proposed in [5].

7.2 Results for bilevel problems with a multi-objective lower level

The set of problems that we tested are bilevel instances where the upper level is a quadratic single-objective problem and the lower level is a multi-objective problem. In particular, given $h_1 \in \mathbb{R}^n$, $h_2 \in \mathbb{R}^m$, a symmetric positive definite matrix $H_2 \in \mathbb{R}^{n \times n}$, and a matrix $H_1 \in \mathbb{R}^{n \times m}$, we solve the general problem

$$\begin{aligned}
\min_{x \in \mathbb{R}^n} f_u(x, y) &= h_1^\top x + h_2^\top y + \frac{1}{2} x^\top H_1 y + \frac{1}{2} x^\top H_2 x, \\
\text{s.t. } y &\in \operatorname{argmin}_{y \in \mathbb{R}^m} F_\ell(x, y) = (f_\ell^1(x, y), f_\ell^2(x, y)),
\end{aligned} \tag{7.1}$$

where the LL objective functions considered in the experiments are specified in Table 1, along with the reference for the LL problem, the number of UL and LL variables (i.e., n and m , respectively), and the bounds on each UL variable x_i . The first two LL objective functions that we consider in our experiments, JOS1 [23] and SP1 [22], are both separable functions, i.e., they can be written as a sum of terms such that each variable only appears in one of the terms. As a result, the third LL objective, which we will refer to as GKV1, leads to a more general multi-objective optimization problem that can be either separable or non-separable depending on the H_1 , H_2 , H_3 , and H_5 matrices that are chosen.

In all the numerical experiments, we considered the same dimension at both the upper and lower levels (i.e., $n = m = \bar{n}$, with \bar{n} positive scalar) and we set H_1 and H_2 in (7.1) equal to identity matrices. The initial points were randomly generated according to a uniform distribution defined within the bounds specified in the last column of Table 1. We compared all the algorithms by using either a line search (LS) or a fixed stepsize (FS) at both the UL and LL problems. We also considered a decaying stepsize but this led to worse performance and, therefore, we

do not report the corresponding results. Recalling the set Λ_{RN} introduced in Section 5, when running BSG-RN on problems with dimension $\bar{n} > 1$, we use $N = 500$ and $Q = 20$ (see Figure 5 for a comparison of the results obtained for different values of Q). When $\bar{n} = 1$, we use $N = Q = 500$. For BSG-OPT and BSG-RN, we implemented an increasing accuracy strategy for the LL problem by using f_u difference thresholds of 0.1 and 0.9, respectively, and a maximum number of LL iterations equal to 30. Note that one could also consider an increasing accuracy strategy for BSG-RA to gradually improve the approximation of the solution of problem (6.5) obtained in Step 1 of Algorithm 4. In this paper, for the practical algorithm considered for BSG-RA, we solve problem (6.5) by using the version of the trust-region method developed in [5] available in the SciPy library [40], with default parameters. In the figures, when comparing the algorithms in terms of iterations, we plot the true function values f_{OPT} and f_{RN} for BSG-OPT and BSG-RN and an accurate approximation of the true function f_{RN} for BSG-RA.

We considered three sets of experiments corresponding to three different settings for the LL problem: deterministic separable case, deterministic non-separable case, and stochastic non-separable case. In the latter case, the UL problem is considered stochastic as well.

Deterministic separable LL case. In this case, we consider LL objective functions that are separable, i.e., all the problems from Table 1. In Problem 3, we set H_3, H_4, H_5 , and H_6 equal to identity matrices and we consider the bounds on x_i given by $[-\infty, 0]$. In (7.1), we set h_1 and h_2 equal to vectors of ones (except for Problem 3, where each element of h_1 is equal to 3). For the problems in this case, we consider $\bar{n} = 1$, which allows us to visualize the solution space in two dimensions for a more direct interpretation of the results.

Figures 1–3 show the results obtained by Algorithms 2–4 when a backtracking Armijo line search [32] at both the UL and LL problems and exact Hessians are used. The UL line search ensures a sufficient decrease of an accurate approximation of the true functions f_{OPT} , f_{RN} , and f_{RA} . We denote the UL optimal solutions found by BSG-OPT, BSG-RN, and BSG-RA as x_{OPT} , x_{RN} , and x_{RA} , respectively. Moreover, we denote the optimal solutions of the problems $\min_{y \in P(x_{\text{OPT}})} f_u(x_{\text{OPT}}, y)$ and $\max_{y \in P(x_{\text{RA}})} f_u(x_{\text{RA}}, y)$ as y_{OPT} and y_{RA} , respectively. In each of these figures, in the upper left-hand plot, we compare the values of the true functions f_{OPT} , f_{RN} , and f_{RA} achieved by each algorithm in terms of iterations. In the upper right-hand plot, we compare the sets $\{(x^*, y) \mid y \in P(x)\}$, where x^* denotes the UL optimal solution determined by each algorithm (i.e., $x^* \in \{x_{\text{OPT}}, x_{\text{RN}}, x_{\text{RA}}\}$), and we also report the contour lines of the UL objective function. In the lower plots, we compare the Pareto fronts between the LL objective functions obtained for each UL optimal solution in $\{x_{\text{OPT}}, x_{\text{RN}}, x_{\text{RA}}\}$, and we refer to the points (f_ℓ^1, f_ℓ^2) evaluated at $(x_{\text{OPT}}, y_{\text{OPT}})$ and $(x_{\text{RA}}, y_{\text{RA}})$ as the optimistic and pessimistic Pareto points, respectively. Note that the optimistic Pareto front *dominates* both the risk-neutral and risk-averse Pareto fronts in all the figures, although the three fronts correspond to different UL variable values. We point out that all the algorithms were able to find the optimal solutions to Problems 1–3.

Deterministic non-separable LL case. In this case, we consider Problem 3 from Table 1 with $\bar{n} = 50$, H_3 and H_5 equal to randomly generated symmetric positive definite matrices, H_4 and H_6 equal to identity matrices, and bounds on x_i given by $[0, \infty]$. In (7.1), the components of the vectors h_1 and h_2 have been randomly generated according to a uniform distribution between -5 and 0 and between -3 and 0 , respectively. The results obtained are shown in Figure 4, where the relative positions of the curves are consistent with the ones in Figures 1–3.

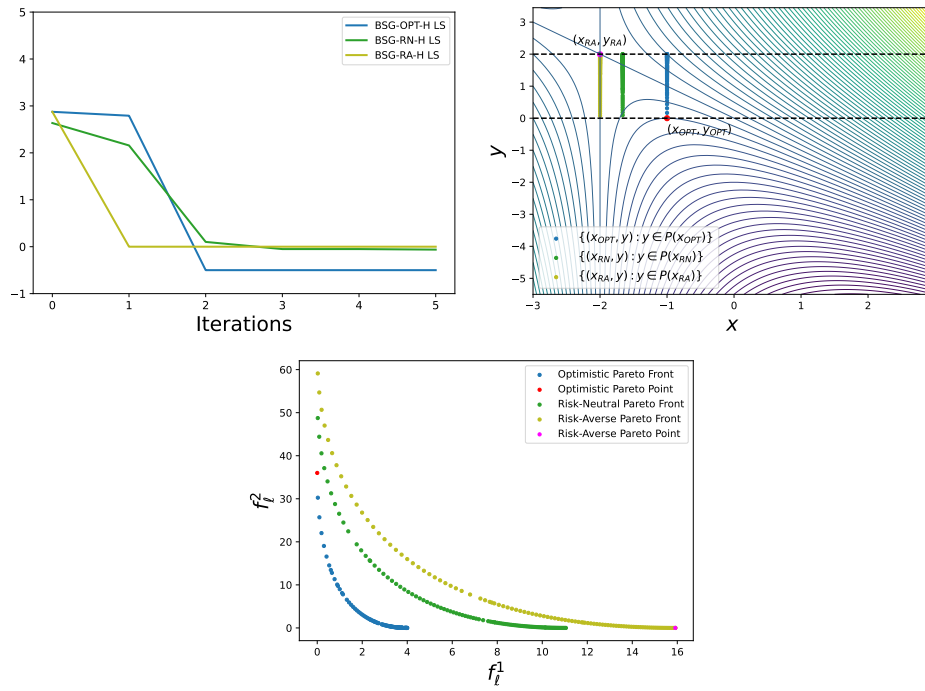


Figure 1: Results for Problem 1 in the deterministic separable LL case (for \bar{n} in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

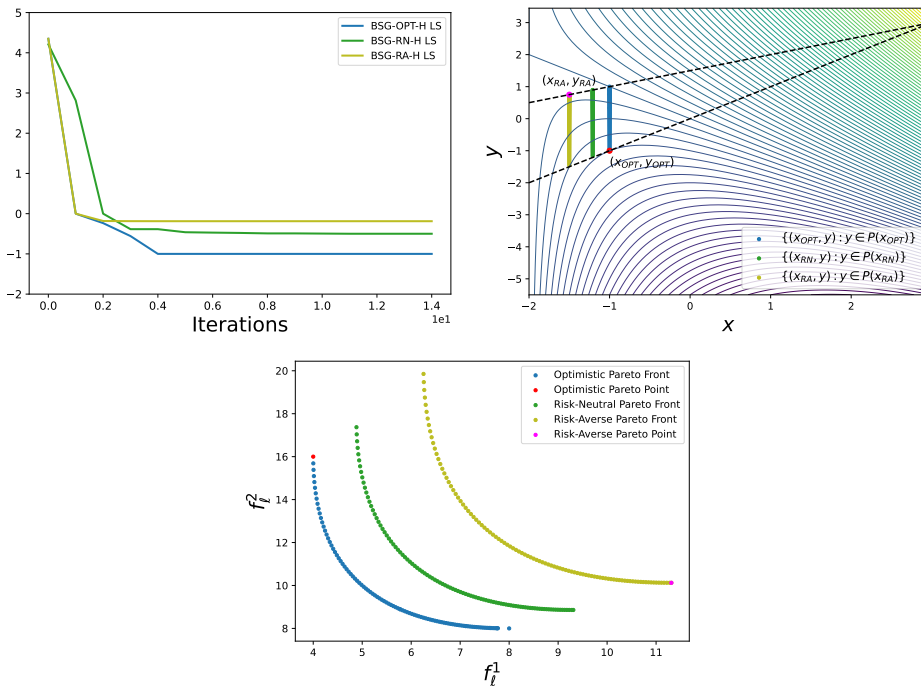


Figure 2: Results for Problem 2 in the deterministic separable LL case (for \bar{n} in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

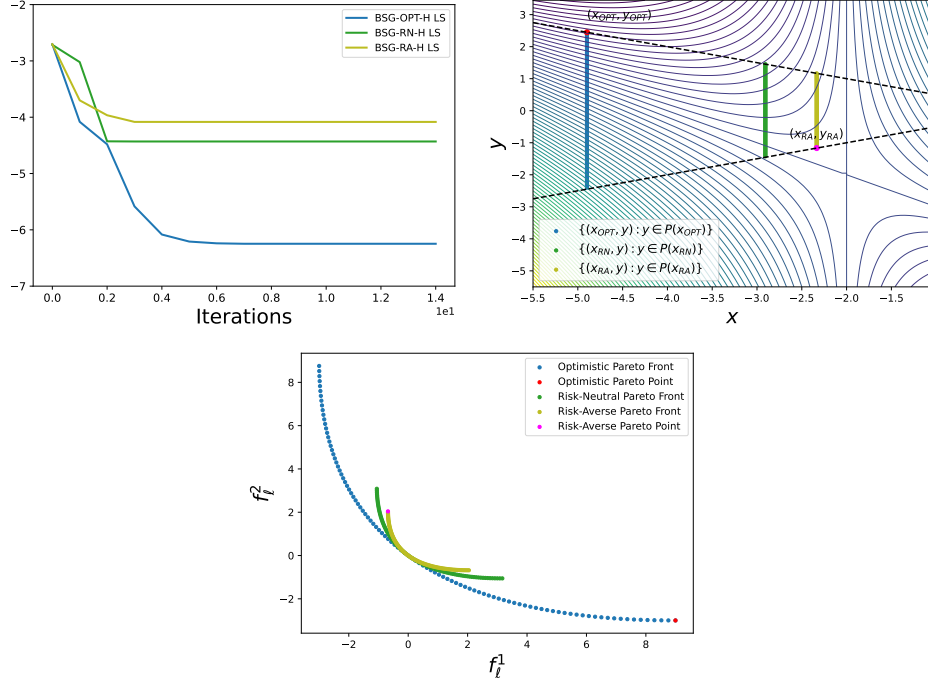


Figure 3: Results for Problem 3 in the deterministic separable LL case (for \bar{n} in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

Figure 5 shows the results when BSG-RN is run with $N = 500$ and $Q \in \{10, 20, 40, 500\}$ in terms of iterations and time. The results in Figure 5 were also obtained by computing the 95% confidence intervals produced over 10 randomly generated starting points. Note that the starting point does not seem to have an impact on the convergence of the algorithms here. Further, one can also see that randomly sampling a set of Q samples from Λ_{RN} leads to the same optimal function value as using the entire set Λ_{RN} and, therefore, confirms the validity of the approach.

Stochastic non-separable LL case. Note that problem (7.1) is deterministic. To investigate the numerical performance of the stochastic methods considered in the experiments, we compute stochastic gradient and Hessian estimates by adding Gaussian noise with mean 0 to each corresponding deterministic gradient (i.e., $\nabla_x f_u$, $\nabla_y f_u$, $\nabla_x f_\ell$, $\nabla_y f_\ell$) and Hessian (i.e., $\nabla_{xy}^2 f_\ell$, $\nabla_{yy}^2 f_\ell$). The values of the noise standard deviation were chosen from the set $\{0, 1, 2\}$ for the gradients and $\{0, 0.1, 0.2\}$ for the Hessians. It is well known that Hessians require larger estimation batches than gradients when considering stochastic settings [2, Section 6.1.1]. We compared all the algorithms by using the best UL and LL stepsizes found for each of them, which were 0.1 for the UL (1 for BSG-RN) and 0.001 for the LL. Such stepsizes were obtained by performing a grid search over the set $\{1, 0.1, 0.01, 0.001\}$ for the UL and the set $\{0.01, 0.001, 0.0001\}$ for the LL. We averaged all the results over 10 trials by using different random seeds and displayed the corresponding 95% confidence intervals. Figure 6 shows that as the value of the standard deviation increases, the performance of all the algorithms gets worse. Further, one can also see that BSG-RN exhibits higher robustness to the noise than BSG-OPT and BSG-RA.

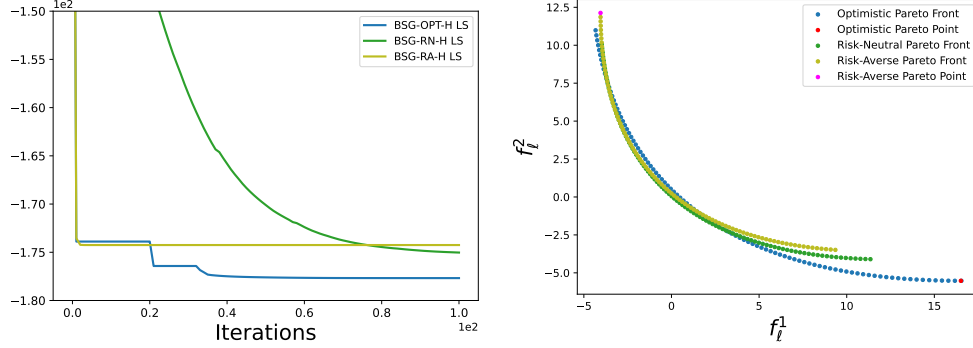


Figure 4: Results for Problem 3 in the deterministic non-separable LL case (for \bar{n} in Table 1 equal to 50). In the left plot, the vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

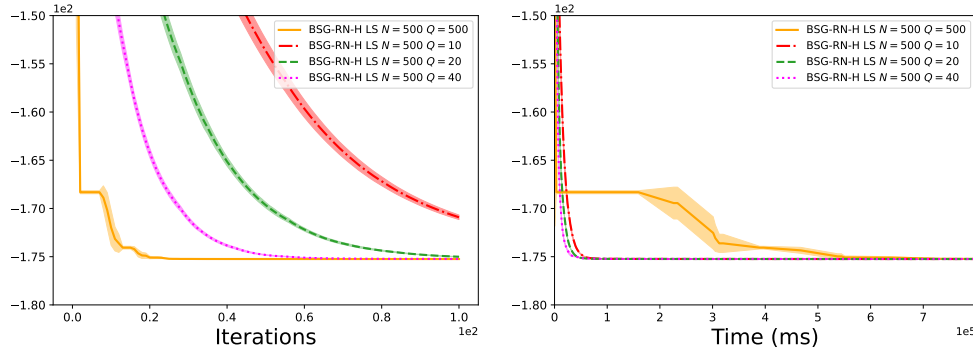


Figure 5: Results for Problem 3 in the deterministic non-separable LL case when running BSG-RN with different values of Q (for \bar{n} in Table 1 equal to 50). In both plots, the vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

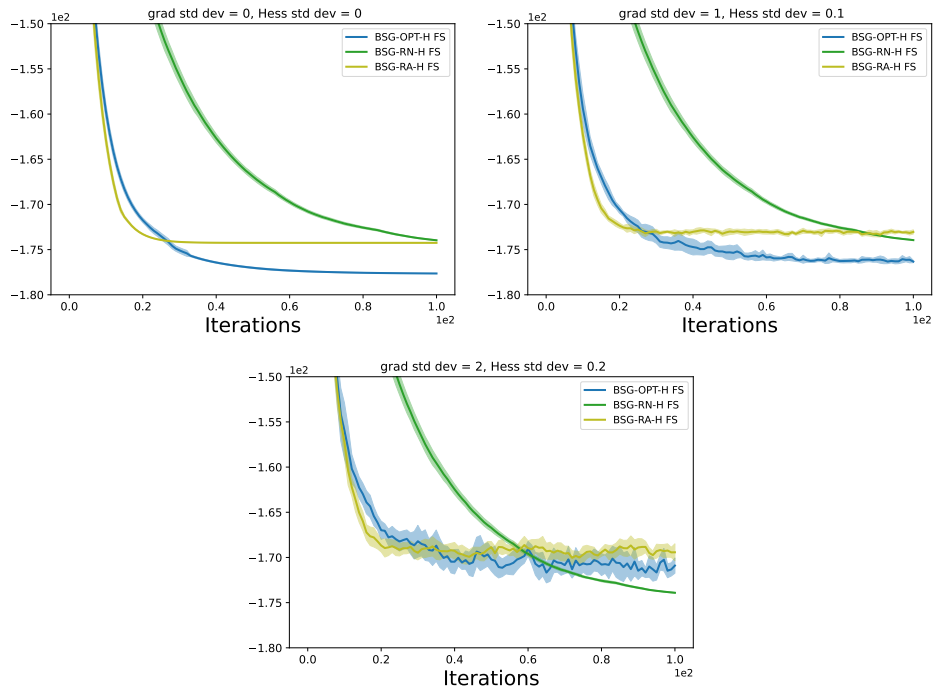


Figure 6: Results for Problem 3 in the stochastic non-separable LL case for different standard deviation values of the stochastic gradient and Hessian estimates (for \bar{n} in Table 1 equal to 50). The vertical axis represents the values of f_{OPT} , f_{RN} , and f_{RA} .

8 Concluding remarks

In this paper we focused on BMOLL problems (upper-level single-objective and lower-level multi-objective) for which we developed new risk-neutral and risk-averse formulations for both the deterministic and stochastic cases and extended the application of the optimistic formulation to the stochastic case. We also developed corresponding gradient-based algorithms.

Other problems that can be considered in bilevel multi-objective optimization are ones with multiple objective functions at the UL problem or at both the UL and LL problems. To address such cases, one can assemble together known approaches from multi-objective and bilevel optimization, as mentioned in Subsections 8.1–8.2 below.

8.1 The multi-objective single-objective case

To develop a gradient-based algorithm for problem (1.1) with $p \geq 1$ and $q = 1$, one can consider the equivalent formulation

$$\min_{x \in \mathbb{R}^n} F_u(x, y(x)) \quad \text{s.t.} \quad x \in X, \quad (8.1)$$

where $y(x)$ denotes the solution of the LL problem. Note that this is essentially a multi-objective problem where the variables y are given by $y(x)$. Therefore, one can draw inspiration from the multi-gradient method developed for deterministic and stochastic multi-objective optimization, for which one knows how to calculate steepest descent directions (i.e., multi-gradients) [11, 16] and stochastic multi-gradients [10, 27]. In (8.1), the resulting multi-gradients are given by *adjoint multi-gradients*. Denoting the current iterate as x and the adjoint gradients of the individual UL objective functions $f^i(x) = f_u^i(x, y(x))$ as $\nabla f^i(x)$, with $i \in \{1, \dots, p\}$, the adjoint multi-gradient can be obtained by first solving the QP subproblem

$$\min_{\delta \in \mathbb{R}^p} \left\| \sum_{i=1}^p \delta_i \nabla f^i(x) \right\|^2 \quad \text{s.t.} \quad \delta \in \Delta, \quad (8.2)$$

where $\Delta = \{\delta \in \mathbb{R}^p : \sum_{i=1}^p \delta_i = 1, \delta_i \geq 0 \forall i \in \{1, \dots, p\}\}$ denotes the simplex set (note that Δ is a subset of \mathbb{R}^p and, therefore, is a different simplex set from Λ introduced in (4.3), which is a subset of \mathbb{R}^q). Then, the negative adjoint multi-gradient is given by $-\sum_{i=1}^p \delta_i^* \nabla f^i(x)$, where δ_i^* is the optimal solution of problem (8.2). In the stochastic case, all the gradients and Hessians in the p adjoint gradients used in problem (8.2) are replaced by their corresponding stochastic estimates.

8.2 The multi-objective multi-objective case

To address problem (1.1) with $p \geq 1$ and $q \geq 1$, one can introduce optimistic, risk-neutral, and risk-averse formulations by following similar approaches to the ones described in Sections 4–6. In particular, one can use an adjoint multi-gradient method to solve the UL problem (see Subsection 8.1) and consider optimistic, risk-neutral, and risk-averse formulations to address the LL problem in the general case $P(x)$. In the optimistic and risk-neutral cases, the resulting algorithms differ from each other in terms of the gradients $\nabla f^i(x)$ used in the QP subproblem (which is (8.2) in the LL single-objective case), for all $i \in \{1, \dots, p\}$. More specifically, in the optimistic case, the vector of weights $\lambda \in \mathbb{R}^q$, which is associated with the LL objective functions,

is included among the UL variables, and the adjoint gradients of the individual objective functions $f_{\text{OPT}}^i(x, \lambda) = f_u^i(x, y(x, \lambda))$ are given by $\nabla f_{\text{OPT}}^i = (\nabla_x f_{\text{OPT}}^i, \nabla_\lambda f_{\text{OPT}}^i)$. In the risk-neutral case, each individual objective function is given by $f_{\text{RN}}^i(x) = (1/N) \sum_{t=1}^N f_u^i(x, y(x, \lambda^t))$, and the resulting gradients are $\nabla f_{\text{RN}}^i(x) = (1/N) \sum_{t=1}^N \nabla f_{\text{RN}}^{it}(x)$, where $f_{\text{RN}}^{it}(x) = f_u^i(x, y(x, \lambda^t))$ for all $i \in \{1, \dots, p\}$ and $t \in \{1, \dots, N\}$. Developing an algorithm for the risk-averse case by following the approach introduced in Section 6 leads to a constrained problem like (6.5) with multiple objective functions. Solving such a problem requires an algorithm for multi-objective constrained problems and, therefore, further research is needed to make the resulting algorithm efficient, robust, and scalable.

Acknowledgments

This work is partially supported by the U.S. Air Force Office of Scientific Research (AFOSR) award FA9550-23-1-0217.

References

- [1] R. Andreani, V. A. Ramirez, S. A. Santos, and L. D. Secchin. Bilevel optimization with a multiobjective problem in the lower level. *Numer. Algorithms*, 81:915–946, jul 2019.
- [2] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60:223–311, 2018.
- [3] C. Chen, X. Chen, C. Ma, Z. Liu, and X. Liu. Gradient-based bi-level optimization for deep learning: A survey. *arXiv preprint arXiv:2207.11719*, 2022.
- [4] T. Chen, Y. Sun, and W. Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25294–25307. Curran Associates, Inc., 2021.
- [5] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, USA, 2000.
- [6] N. Couellan and W. Wang. Bi-level stochastic gradient for large scale support vector machine. *Neurocomputing*, 153:300–308, 2015.
- [7] N. Couellan and W. Wang. On the convergence of stochastic bi-level gradient methods. *Preprint available at http://www.optimization-online.org/DB_HTML/2016/02/5323.html*, 2016.
- [8] J.M. Danskin. *The Theory of Max-Min and its Application to Weapons Allocation Problems*. Econometrics and Operations Research. Springer, 1967.
- [9] S. Dempe and P. Mehrlitz. Semivectorial bilevel programming versus scalar bilevel programming. *Optimization*, 69:657–679, 2020.
- [10] J. A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *C. R. Math. Acad. Sci. Paris*, 350:313–318, 2012.

- [11] L. G. Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175:395–414, 2005.
- [12] M. Ehrgott. *Multicriteria Optimization*, volume 491. Springer Science & Business Media, Berlin, 2005.
- [13] G. Eichfelder. Multiobjective bilevel optimization. *Math. Program.*, 123:419–449, jun 2010.
- [14] G. Eichfelder. *Methods for multiobjective bilevel optimization*, pages 423–449. Springer International Publishing, Cham, 2020.
- [15] A. V. Fiacco. Optimal value differential stability bounds under the mangasarian-fromovitz constraint qualification. *Mathematical Programming with Data Perturbations*, 2:65–90, 1983.
- [16] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.*, 51:479–494, 2000.
- [17] A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.*, 22:618–630, 1968.
- [18] S. Ghadimi and M. Wang. Approximation methods for bilevel programming. *arXiv e-prints*, art. arXiv:1802.02246, 2018.
- [19] T. Giovannelli, G. Kent, and L. N. Vicente. Inexact bilevel stochastic gradient methods for constrained and unconstrained lower-level problems. *ISE Technical Report 21T-025, Lehigh University*, December 2022.
- [20] A. Gu, S. Lu, P. Ram, and L. Weng. Min-max bilevel multi-objective optimization with applications in machine learning. *arXiv e-prints*, art. arXiv:2203.01924, March 2022.
- [21] M. Hong, H. Wai, Z. Wang, and Z. Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv e-prints*, art. arXiv:2007.05170, July 2020.
- [22] S. Huband, P. Hingston, L. Barone, and R. Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10:477–506, 2006.
- [23] Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? GECCO’01, page 1042–1049, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [24] M. M. Kamani, S. Farhang, M. Mahdavi, and J. Z. Wang. Targeted data-driven regularization for out-of-distribution generalization. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 882–891, New York, NY, USA, 2020. Association for Computing Machinery.
- [25] A. M. Lessin, B. J. Lunday, and R. R. Hill. A multi-objective, bilevel sensor relocation problem for border security. *IIEE Transactions*, 51:1091–1109, 2019.

- [26] R. Liu, J. Gao, J. Zhang, D. Meng, and Z. Lin. Investigating bi-Level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv e-prints*, art. arXiv:2101.11517, January 2021.
- [27] S. Liu and L. N. Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pages 1–30, 2021.
- [28] X. Luo, Y. Liu, and X. Liu. Bi-level multi-objective optimization of design and subsidies for standalone hybrid renewable energy systems: A novel approach based on artificial neural network. *Journal of Building Engineering*, 41:102744, 2021.
- [29] Y. Lv and Z. Wan. A solution method for the optimistic linear semivectorial bilevel optimization problem. *Journal of Inequalities and Applications*, 2014:164, 05 2014.
- [30] Y. Lü and Z. Wan. A smoothing method for solving bilevel multiobjective programming problems. *Journal of the Operations Research Society of China*, 2:511–525, 12 2014.
- [31] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, New York, 2012.
- [32] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, second edition, 2006.
- [33] M. S. Ozdayi, M. Kantarcioglu, and R. Iyer. BiFair: Training fair models with bilevel optimization. *arXiv e-prints*, art. arXiv:2106.04757, June 2021.
- [34] Y. Roh, K. Lee, S. Euijong Whang, and C. Suh. FairBatch: Batch selection for model fairness. *arXiv e-prints*, art. arXiv:2012.01696, December 2020.
- [35] W. Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Company, Inc., New York-Toronto-London, 1953.
- [36] X. Shi and H.S. Xia. Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers. *Journal of Multi-Criteria Decision Analysis*, 10:27–34, 2001.
- [37] A. Sinha and K. Deb. Towards understanding evolutionary bilevel multi-objective optimization algorithm. *IFAC Proceedings Volumes*, 42:338–343, 2009.
- [38] A. Sinha, P. Malo, and K. Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22:276–295, 2018.
- [39] D. Sow, K. Ji, and Y. Liang. On the convergence theory for Hessian-free bilevel algorithms. *arXiv e-prints*, art. arXiv:2110.07004, October 2021.
- [40] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, and K. Jarrod et al. Millman. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.