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

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

February 10, 2023

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

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

The upper-level (UL) and lower-level (LL) objective functions $F_u : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $F_\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ are given by $F_u = (f_u^1, \ldots, f_u^p)$ and $F_\ell = (f_\ell^1, \ldots, f_\ell^q)$, with f_u^i and f_ℓ^j real-valued functions for all $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q\}$, 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

*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, USA (lnv@lehigh.edu). Support for this author was partially provided by the Centre for Mathematics of the University of Coimbra under grant FCT/MCTES UIDB/MAT/00324/2020.

variables y are subjected to LL constraints ($y \in Y(x)$); however, for the rest of this paper, it is assumed that the LL variables are unconstrained (i.e., $Y(x) = \mathbb{R}^m$). 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 [12, 13, 33]. When a multi-objective problem only arises in one of the two levels, one also finds the terminology semivectorial bilevel optimization [8].

Bilevel multi-objective optimization problems arise in applications related to defense, renewable energy systems, and fair machine learning. In the defense sector, a multi-objective bilevel problem for facility location was proposed in [23] 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 [26], 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 [22, 30, 31], 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

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

where the UL objective function $f_u : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is real-valued. Since the LL problem has conflicting objectives, a single optimal solution must be determined among the set of LL Pareto optimal solutions, and several criteria can be considered that potentially lead to different solutions to 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 [3, 5, 6, 16, 17, 19, 34] (see also [2, 17, 24] 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, 8, 27, 28], 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 [11]. 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 [32], which uses the ε-constraint method [11] 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 [18], where multiple objectives are considered at the upper level. We point

out that the problem solved in [18] is significantly different from problem (1.1) since in [18] there are multiple LL problems and each UL objective function only depends on the optimal solution of one LL problem. Moreover, in [18], 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, addressing their computational challenges and developing deterministic and stochastic gradient descent algorithms. Both formulations are inspired from 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 all x-dependent efficient solution sets. 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-adverse 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). We describe the known optimistic formulation for (1.2) in Section 3. The new risk-neutral and risk-averse formulations for (1.2), as well as the corresponding gradient-based algorithms, are introduced in Sections 4 and 5, respectively. Numerical results for synthetic bilevel problems with a multi-objective lower level are reported in Section 6, which also describes the practical implementations of the proposed methods. Finally, we draw some concluding remarks in Section 7, 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

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, y\in\mathbb{R}^m} \quad & f_u(x,y) \\ \text{s.t.} \quad & x \in X \\ & y \in \operatorname*{argmin}_{y\in\mathbb{R}^m} \quad f_\ell(x,y). \end{aligned} \tag{2.1}$$

In BO problems, the goal of the UL problem is to minimize the UL objective function $f_u : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ 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 $f_\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ over the LL variables y. Whenever orthogonal projections are required by the methods considered in this paper, the set X will be assumed closed and convex.

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,

$$\min_{x\in\mathbb{R}^n} \; f(x) = f_u(x, y(x)) \quad \text{s.t.} \quad x \in X. \tag{2.2}$$

Considering appropriate smoothness and non-singularity assumptions, the gradient of f at x can be obtained from the well-known adjoint (or hypergradient) formula

$$\nabla f \;=\; \nabla_x f_u - \nabla^2_{xy} f_\ell \nabla^2_{yy} f_\ell^{-1} \nabla_y f_u, \tag{2.3}$$

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 resulting equation is given by $\nabla^2_{yx} f_\ell + \nabla^2_{yy} f_\ell \nabla y^\top = 0$ (where all gradients and Hessians are evaluated at $(x, y(x))$), which leads to $\nabla y = -\nabla^2_{xy} f_\ell \nabla^2_{yy} f_\ell^{-1}$.

In Algorithm 1, we report a general framework for the bilevel stochastic gradient (BSG) method [17] 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 approximately solve 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, 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).

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 $k = 0, 1, 2, \ldots$ **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

The convergence theory for the BSG method developed in [17] 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 has been derived in [17] under the assumption of non-convexity, strong convexity, and convexity of the true objective function f. The convergence theory of the algorithms introduced in this paper could be obtained as an extension of the convergence results presented in [17].

3 The known optimistic formulation

Since the LL of problem (1.2) is multi-objective, the set of LL optimal solutions is given by the set of Pareto minimizers (or non-dominated points) of the LL problem, which is denoted by $P(x)$. The resulting so-called optimistic formulation corresponds to the problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, y\in\mathbb{R}^m} \quad & f_u(x, y) \\ \text{s.t.} \quad & x \in X \\ & y \in P(x). \end{aligned} \tag{3.1}$$

When the LL objective functions $f_\ell^1, \ldots, f_\ell^q$ are convex, problem (3.1) is equivalent to

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, \lambda\in\mathbb{R}^q,\, y\in\mathbb{R}^m} \quad & f_u(x,y) \\ \text{s.t.} \quad & x \in X,\ \lambda \in \Lambda \\ & y \in \operatorname*{argmin}_{y\in\mathbb{R}^m} \ f_\ell(x,\lambda,y) := \lambda^\top F_\ell(x,y), \end{aligned} \tag{3.2}$$

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

$$\Lambda = \left\{ \lambda \in \mathbb{R}^q \;\middle|\; \sum_{j=1}^q \lambda_j = 1,\ \lambda_j \geq 0\ \forall j \in \{1,\ldots,q\} \right\}. \tag{3.3}$$

The convexity assumption on the LL objective functions ensures the existence of a vector of weights in Λ for each LL Pareto minimizer in $P(x)$ [11].

Note that the LL objective function in problem (3.2) is $f_\ell(x,\lambda,y) = \sum_{j=1}^q \lambda_j f_\ell^j(x,y)$, where $\lambda \in \Lambda$. Hence, problem (3.2) 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). Denoting the optimal solution of the LL problem in (3.2) by $y(x,\lambda)$ and assuming its uniqueness, let $f_{\rm OPT}(x,\lambda) = f_u(x,y(x,\lambda))$, with OPT standing for optimistic. By applying the chain rule to $f_u(x,y(x,\lambda))$, one obtains the gradient vectors

$$\nabla_x f_{\rm OPT} \ = \ \nabla_x f_u + \nabla_x y \nabla_y f_u, \qquad \nabla_\lambda f_{\rm OPT} \ = \ \nabla_\lambda y \nabla_y f_u. \tag{3.4}$$

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

$$\nabla^2_{yx} f_\ell + \nabla^2_{yy} f_\ell \nabla_x y^\top \ = \ 0, \qquad \nabla^2_{y\lambda} f_\ell + \nabla^2_{yy} f_\ell \nabla_\lambda y^\top \ = \ 0. \tag{3.5}$$

Under the non-singularity assumption of $\nabla^2_{yy} f_\ell$, we can obtain $\nabla_x y$ and $\nabla_\lambda y$ from (3.5) and plug their values into (3.4), which leads to

$$\nabla_x f_{\rm OPT} \ = \ \nabla_x f_u - \nabla^2_{xy} f_\ell (\nabla^2_{yy} f_\ell)^{-1} \nabla_y f_u, \qquad \nabla_\lambda f_{\rm OPT} \ = \ -\nabla^2_{\lambda y} f_\ell (\nabla^2_{yy} f_\ell)^{-1} \nabla_y f_u, \tag{3.6}$$

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

$$\nabla^2_{xy} f_\ell = \sum_{j=1}^q \lambda_j \nabla^2_{xy} f_\ell^j, \quad \nabla^2_{yy} f_\ell = \sum_{j=1}^q \lambda_j \nabla^2_{yy} f_\ell^j, \quad \nabla^2_{\lambda y} f_\ell = \left(\nabla_y f_\ell^1, \ldots, \nabla_y f_\ell^q\right)^\top. \tag{3.7}$$

Therefore, from (3.6)–(3.7), one can obtain the adjoint gradient $\nabla f_{\rm OPT}$ by concatenating the subvectors $\nabla_x f_{\rm OPT}$ and $\nabla_\lambda f_{\rm OPT}$ into a single vector. In the stochastic case, all the gradients and Hessians on the right-hand sides of (3.6) can be replaced by corresponding stochastic estimates.

In Algorithm 2, we introduce a bilevel stochastic gradient method to solve the optimistic interpretation of problem (1.2), which is given by problem (3.2) when the LL objective functions $f_\ell^1, \ldots, f_\ell^q$ are convex. 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 $k = 0, 1, 2, \ldots$ **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 interpretation (3.2) of (1.2), 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 alternative approaches to the optimistic formulation (3.2). In the remainder of this paper, when considering weighted combinations of the LL objective functions, we will assume that the functions f_ℓ^j are convex, for all $j \in \{1, \ldots, q\}$.

4 A new risk-neutral formulation

In this section, we introduce a new formulation for problem (1.2). To gain intuition, suppose that the Pareto front of the LL 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

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

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

$$\min_{x\in X} \quad \mathbb{E}_{y\sim P}\left[f_u(x, y)\right], \tag{4.2}$$

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 (4.2) 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, given x and assuming that y is a random vector with a probability distribution defined over $P(x)$, the problem to solve is

$$\min_{x\in X} \quad \mathbb{E}_{y\sim P(x)}\left[f_u(x, y)\right]. \tag{4.3}$$

Under the assumption on the convexity of the LL objective functions $f_\ell^1, \ldots, f_\ell^q$ introduced in Section 3, we consider a companion problem to (4.3) 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 (3.3)), the problem we will consider instead is

$$\min_{x \in X} \quad \mathbb{E}_{\lambda \sim \Lambda} \left[f_u(x, y_\lambda(x)) \right], \tag{4.4}$$

where $y_\lambda(x)$ is the optimal solution of the problem

$$\min_{y \in \mathbb{R}^m} \quad F_\ell(x, y)^\top \lambda. \tag{4.5}$$

In practice, one can consider a finite set $\Lambda_{\text{RN}} = \{\lambda^1, \ldots, \lambda^N\} \subset \Lambda$ that corresponds to a fine-scale discretization of Λ, with RN standing for risk-neutral. The corresponding set of LL Pareto minimizers is $\{y_{\lambda^1}(x), \ldots, y_{\lambda^N}(x)\} \subset P(x)$. Therefore, problem (4.4) can be formulated as

$$\min_{x \in X} \quad f_{\text{RN}}(x) \;=\; \frac{1}{N} \sum_{i=1}^{N} f_u \left(x, y_{\lambda^i}(x) \right). \tag{4.6}$$

Note that the gradient of the objective function in (4.6) is given by

$$\nabla f_{\text{RN}}(x) \;=\; \frac{1}{N} \sum_{i=1}^{N} \nabla f_{\text{RN}}^i(x), \tag{4.7}$$

where $f_{\text{RN}}^i(x) = f_u(x, y_{\lambda^i}(x))$ for all $i \in \{1, \ldots, N\}$. By applying the chain rule to $f_u(x, y_{\lambda^i}(x))$, one obtains

$$\nabla f_{\text{RN}}^i \;=\; \nabla_x f_u + \nabla y_{\lambda^i} \nabla_y f_u. \tag{4.8}$$

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

$$\sum_{j=1}^{q} \lambda_j^i \left(\nabla_{yx}^2 f_\ell^j + \nabla_{yy}^2 f_\ell^j \nabla y_{\lambda^i}^\top \right) = 0, \tag{4.9}$$

where all Hessians are evaluated at $(x, y_{\lambda_i}(x))$. Equation (4.9) yields

$$\nabla y_{\lambda^i} \;=\; -\left(\sum_{j=1}^{q} \lambda_j^i \nabla_{xy}^2 f_\ell^j \right) \left(\sum_{j=1}^{q} \lambda_j^i \nabla_{yy}^2 f_\ell^j \right)^{-1}, \text{ for all } i \in \{1, \ldots, N\}. \tag{4.10}$$

Plugging (4.10) into (4.8), one obtains

$$\nabla f_{\text{RN}}^i \;=\; \nabla_x f_u - \left(\sum_{j=1}^{q} \lambda_j^i \nabla_{xy}^2 f_\ell^j \right) \left(\sum_{j=1}^{q} \lambda_j^i \nabla_{yy}^2 f_\ell^j \right)^{-1} \nabla_y f_u, \tag{4.11}$$

where all gradients and Hessians on the right-hand side are evaluated at $(x, y_{\lambda^i}(x))$. In the stochastic case, all the gradients and Hessians on the right-hand side of (4.11) 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 (4.6) by randomly choosing a set of samples (i.e., a mini-batch) from Λ_{RN}. Denoting a mini-batch as $\Lambda_Q = \{\lambda^1, \ldots, \lambda^Q\} \subseteq \Lambda_{\mathrm{RN}}$, where Q is the mini-batch size, the corresponding sample set of LL Pareto minimizers $\{y_{\lambda^1}(x), \ldots, y_{\lambda^Q}(x)\}$ can be used to compute a SG for ∇f_{RN} in (4.7).

In Algorithm 3, we introduce a bilevel stochastic gradient method to solve the risk-neutral interpretation of problem (1.2) given by the formulation (4.6). 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 $i \in \{1, \ldots, Q\}$. For the sake of simplicity, we denote $\tilde{y}_k = \{(\tilde{y}_{\lambda^i})_k \mid i \in \{1, \ldots, Q\}\}$ and $\xi_k = \{(\xi_i)_k \mid i \in \{1, \ldots, Q\}\}$.

Algorithm 3 BSG-RN Method

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

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

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

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

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

End do

5 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 (4.1). The *risk-averse* (or robust or pessimistic) formulation is given by the problem

$$\min_{x\in X} \max_{y\in P} \; f_u(x, y), \tag{5.1}$$

which can be reformulated as $\min_{x\in X} \; f_{\mathrm{RA}}(x)$ by introducing $f_{\mathrm{RA}}(x) = \max_{y\in P} \; f_u(x, y)$, 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 [7], such a subdifferential is given by $\partial f_{\mathrm{RA}}(x) = \mathrm{conv}\{\nabla_x f_u(x, y) \mid y \in Y_0(x)\}$, 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, the problem to solve is

$$\min_{x\in X} \max_{y\in P(x)} \; f_u(x, y), \tag{5.2}$$

which can be reformulated as $\min_{x\in X} \; f_{\mathrm{RA}}(x)$ by introducing $f_{\mathrm{RA}}(x) = \max_{y\in P(x)} \; f_u(x, y)$. Let us specify $P(x)$ algebraically by using the LL first-order necessary conditions for Pareto

optimality as follows

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

Under the assumption of the convexity of the LL objective functions $f_\ell^1, \ldots, f_\ell^q$ introduced in Section 3, problem (5.2) is equivalent to the problem

$$\min_{x \in X} \max_{(y,\lambda) \in \tilde{P}(x)} f_u(x,y), \tag{5.3}$$

where

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

Problem (5.3) can be reformulated as $\min_{x \in X} \; \tilde{f}_{\mathrm{RA}}(x)$ by introducing $\tilde{f}_{\mathrm{RA}}(x) = \max_{(y,\lambda) \in \tilde{P}(x)} \; f_u(x,y)$. The optimization problem defining $\tilde{f}_{\mathrm{RA}}(x)$ can be written as

$$\begin{aligned} \max_{y \in \mathbb{R}^m,\, \lambda \in \mathbb{R}^q} \quad & f_u(x,y) \\ \text{s.t.} \quad & \lambda \in \Lambda, \\ & \sum_{j=1}^{q} \lambda_j \nabla_y f_\ell^j(x,y) = 0. \end{aligned} \tag{5.4}$$

The Lagrangian function of problem (5.4) is $\mathcal{L}(x,y,\lambda,z) = f_u(x,y) + z_I^\top \lambda + z_E^\top (\sum_{i=1}^q \lambda_i - 1, \sum_{j=1}^q \lambda_j \nabla_y f_\ell^j(x,y))$, 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. Based on [14, Corollary 4.11], under the assumption that the gradients of the active constraints are linearly independent, the subdifferential of $\tilde{f}_{\mathrm{RA}}$ at x is now given by

$$\partial \tilde{f}_{\mathrm{RA}}(x) \;=\; \mathrm{conv}\{\nabla_x \mathcal{L}(x, y(x), \lambda(x), z(x, y(x), \lambda(x))) \mid (y(x), \lambda(x)) \in \tilde{Y}_0(x)\}, \tag{5.5}$$

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. Note that the gradient of the Lagrangian with respect to x is given by

$$\nabla_x \mathcal{L}(x,y,\lambda,z) \;=\; \nabla_x f_u(x,y) + \sum_{j=1}^{q} \lambda_j \nabla_{xy}^2 f_\ell^j(x,y) z_E. \tag{5.6}$$

Equation (5.6) can be used in (5.5), where $\tilde{Y}_0(x)$ is given by the solutions $(\bar{y}(x), \bar{\lambda}(x))$ of problem (5.4). In the stochastic case, all the gradients and Hessians on the right-hand side of (5.5)–(5.6) can be replaced by corresponding stochastic estimates. When using stochastic estimates, the subdifferential (5.5) is denoted as $\partial \tilde{f}_{\mathrm{RA}}^d(x)$.

In Algorithm 4, we introduce a bilevel stochastic subgradient method to solve the risk-averse interpretation of problem (1.2) given by problem (5.2). 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 $k = 0, 1, 2, \ldots$ **do**

Step 1. Obtain an approximation $(\tilde{y}_k, \tilde{\lambda}_k)$ to a solution $(y(x_k), \lambda(x_k))$ of problem (5.4).

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

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

6 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).

6.1 Our practical methods

A major difficulty in (3.6), (4.11), and (5.6) is the use of second-order derivatives of f^j_ℓ, which prevents the application of Algorithms 2–4 to large-scale problems. In the practical algorithms that we propose, we avoid this problem by approximating the second-order derivatives with the outer products of the corresponding gradients, which have successfully been applied in [17] for Algorithm 1 and are inspired by Gauss-Newton methods for nonlinear least-squares problems. Such rank-1 approximations are as follows:

$$\nabla^2_{xy} f^j_\ell \;\simeq\; \nabla_x f^j_\ell \nabla_y f^{j\top}_\ell \;\text{ and }\; \nabla^2_{yy} f^j_\ell \;\simeq\; \nabla_y f^j_\ell \nabla_y f^{j\top}_\ell, \text{ for all } \; j \in \{1, \ldots, q\}. \tag{6.1}$$

When using these rank-1 approximations, Algorithms 2–4 will be referred to as BSG-OPT-1, BSG-RN-1, and BSG-RA-1, respectively, where the "1" stands for first-order rank-1 approximations of the Hessian matrices. Due to the small dimensions of the problems that we will be testing $(n, m \leq 50)$, in BSG-OPT and BSG-RN, the resulting adjoint systems are solved by factorizing the matrices $\sum_{j=1}^q \lambda_j \nabla^2_{yy} f^j_\ell$ and $\sum_{j=1}^q \lambda^i_j \nabla^2_{yy} f^j_\ell$, respectively. However, when the dimensions of the problems are large, one can solve the resulting adjoint systems via the linear conjugate gradient method until non-positive curvature is detected. We will also test Algorithms 2–4 with both exact and stochastic Hessians for the deterministic and stochastic settings, respectively. These versions are referred to as BSG-OPT-H, BSG-RN-H, and BSG-RA-H. Regardless of the way used to handle the second-order derivatives, we apply either gradient descent (in the deterministic setting) or stochastic gradient descent (in the stochastic setting) for a certain budget of iterations in Step 1 of BSG-OPT and BSG-RN. 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 [17]. In Step 1 of BSG-RA, we solve problem (5.4) by applying the trust-region algorithm for nonlinear constrained problems proposed in [4].

6.2 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 $h_1 \in$

Problem	n	m	Ref. for LL	LL Objective Functions	Bound on x_i
1	$\bar{n}$	$\bar{n}$	JOS1 [21]	$f_\ell^1(x,y) = \frac{1}{\bar{n}}\sum_{i=1}^{\bar{n}} x_i^2 y_i^2$ $f_\ell^2(x,y) = \frac{1}{\bar{n}}\sum_{i=1}^{\bar{n}} (x_i-2)^2(y_i-2)^2$	$[-2,\infty]$
2	$\bar{n}$	$\bar{n}$	SP1 [20]	$f_\ell^1(x,y) = \sum_{i=1}^{\bar{n}}[(x_i-1)^2+(x_i-y_i)^2]$ $f_\ell^2(x,y) = \sum_{i=1}^{\bar{n}}[(y_i-3)^2+(x_i-y_i)^2]$	$[-2,3]$
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).

$\mathbb{R}^n$, $h_2 \in \mathbb{R}^m$, 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

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \; & f_u(x,y) \;=\; h_1^\top x + h_2^\top y + \frac{1}{2}x^\top H_1 y + \frac{1}{2}x^\top H_2 x, \\ & y \in \operatorname*{argmin}_{y\in\mathbb{R}^m} \; F_\ell(x,y) = (f_\ell^1(x,y), f_\ell^2(x,y)), \end{aligned} \tag{6.2}$$

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 [21] and SP1 [20], respectively, are both separable functions, i.e., they can be written as the 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, is a more general MO 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 (6.2) 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 results and, therefore, we do not report the corresponding results. Recalling the set Λ_{RN} introduced in Section 4, 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 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 (5.4) obtained in Step 1 of Algorithm 4. In this paper, for the practical algorithm considered for BSG-RA, we solve problem (5.4) by using the version of the trust-region method developed in [4] available in the SciPy library [35], 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 (6.2), 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 [29] at both the UL and LL problems and true 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 $\min_{y \in P(x_{\text{OPT}})} f_u(x, y)$ and $\max_{y \in P(x_{\text{RA}})} f_u(x, y)$ 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 plot, 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 $[0, \infty]$. For this case, we also compare the performance of the algorithms when using the rank-1 approximations versus the true Hessians. In (6.2), 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 one can see that the algorithms that use rank-1 approximations achieve the same objective function value as the corresponding algorithms using true Hessians. Figure 5 shows the results obtained by running BSG-RN 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 (6.2) is deterministic. To investigate the numerical performance of the stochastic methods considered in the experiments, we

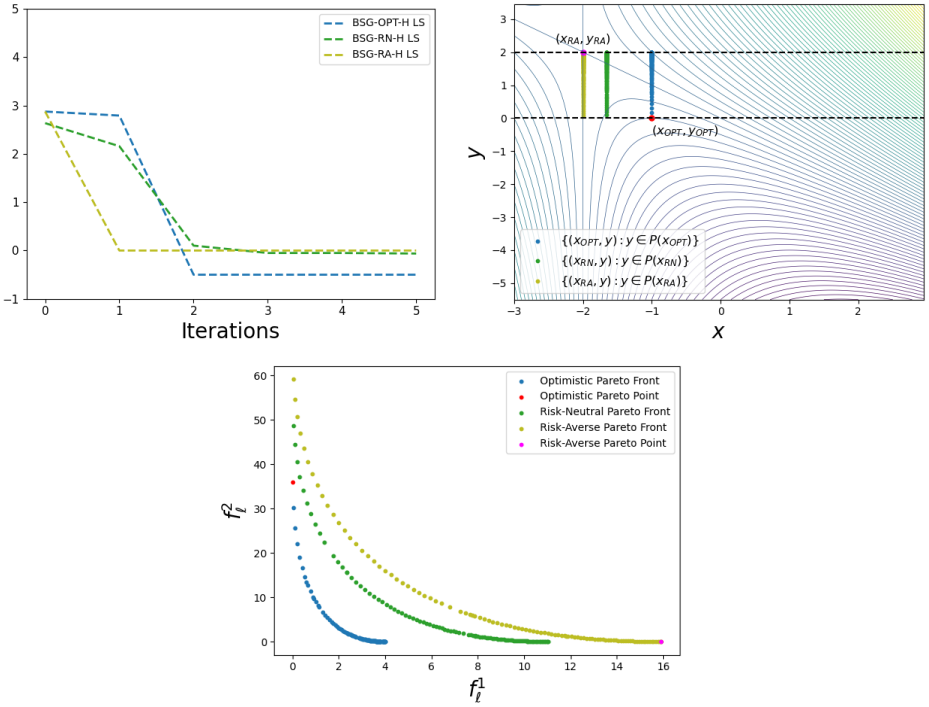

Figure 1: Results for Problem 1 in the deterministic separable LL case (for $\bar{n}$ in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

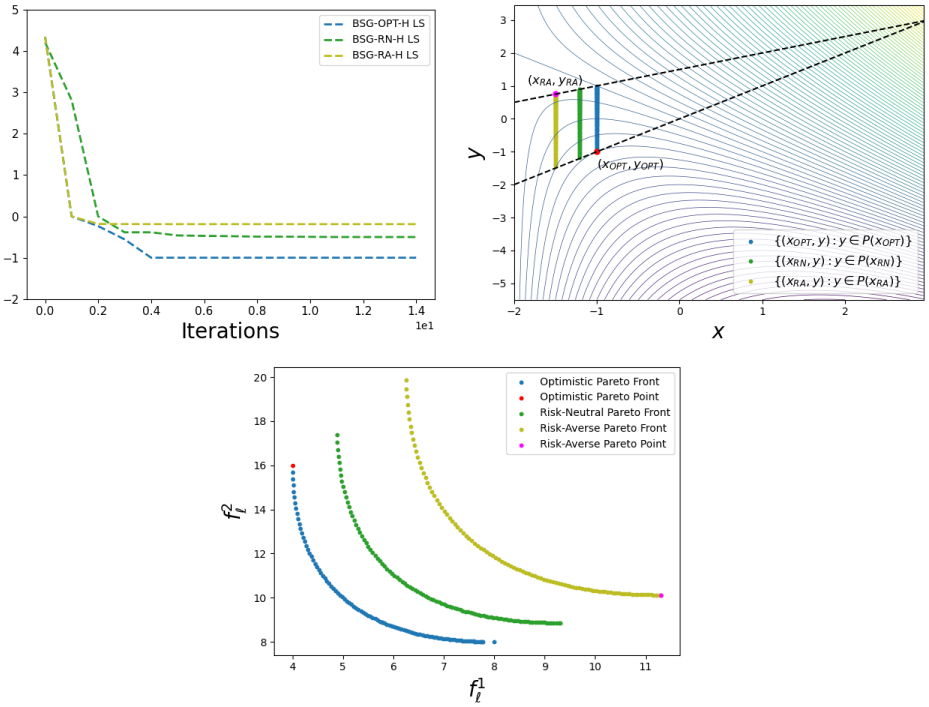

Figure 2: Results for Problem 2 in the deterministic separable LL case (for $\bar{n}$ in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

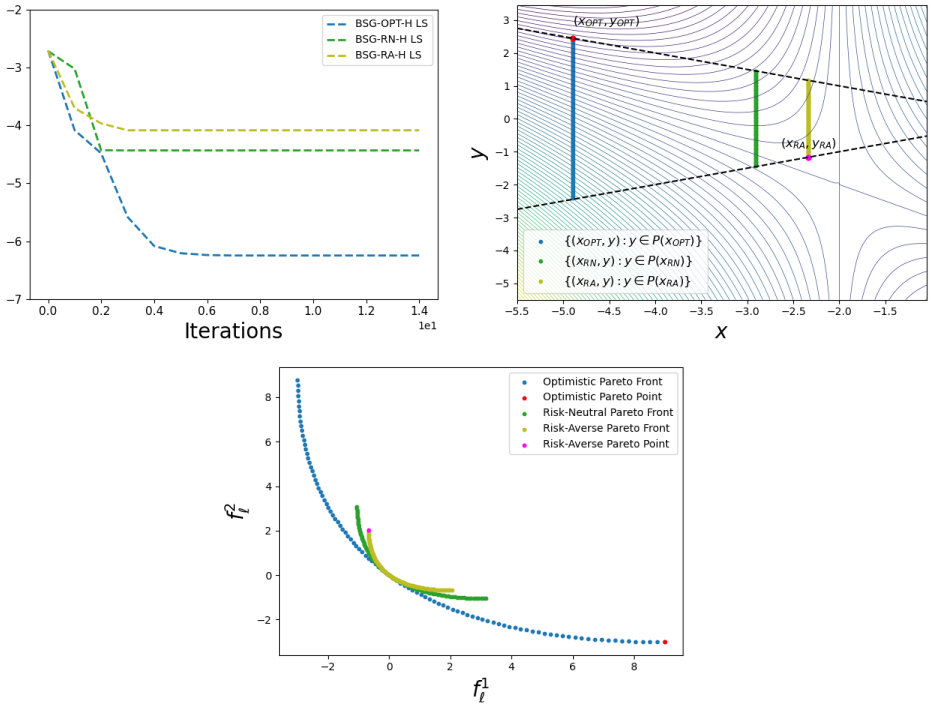

Figure 3: Results for Problem 3 in the deterministic separable LL case (for $\bar{n}$ in Table 1 equal to 1). In the upper-left plot, the vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

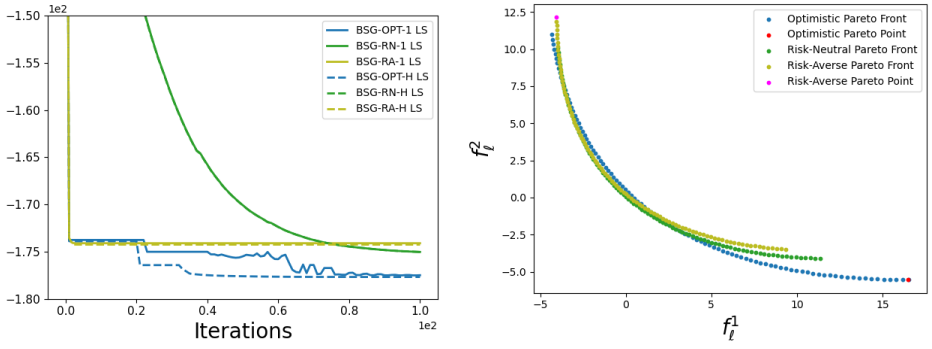

Figure 4: Results for Problem 3 in the deterministic non-separable LL case (for $\bar{n}$ in Table 1 equal to 50). In the left plot, the vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

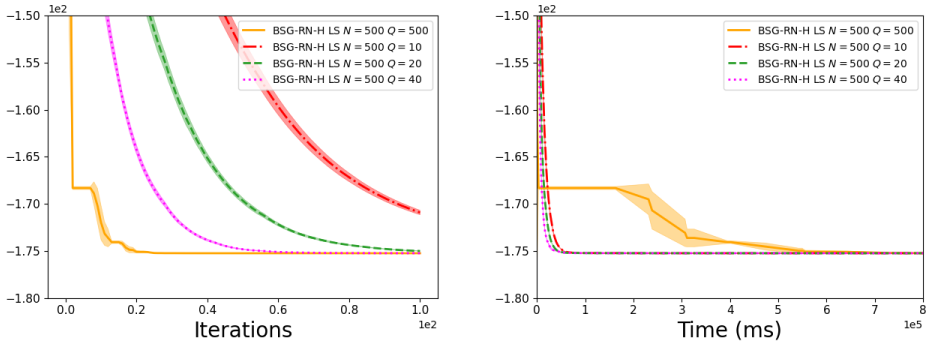

Figure 5: Results for Problem 3 in the deterministic non-separable LL case when running BSG-RN with different values of Q (for $\bar{n}$ in Table 1 equal to 50). In both plots, the vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

compute stochastic gradient and Hessian estimates by adding Gaussian noise with mean 0 to each corresponding deterministic gradients (i.e., $\nabla_x f_u$, $\nabla_y f_u$, $\nabla_x f_\ell$, $\nabla_y f_\ell$) and Hessians (i.e., $\nabla^2_{xy} f_\ell$, $\nabla^2_{yy} f_\ell$). In this case, we again compare the performance using the rank-1 approximations versus the true Hessians. The values of the standard deviation were chosen from the set $\{0, 0.05, 0.1, 0.5, 1, 2\}$. 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 display 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. We omit the algorithms from the plots when we are not able to find stepsizes that ensure performance comparable to the other algorithms in the plots. One can see that BSG-OPT and BSG-RN exhibit higher robustness to the noise than BSG-RA, which yields objective function values that start increasing after a few iterations even for small values of the standard deviation. In general, as the value of the standard deviation increases, the use of rank-1 approximations guarantees better performance than the counterparts that use true Hessians.

7 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 and proposed the use of rank-1 approximations that allow applying such algorithms to large-scale problems.

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 7.1–7.2.

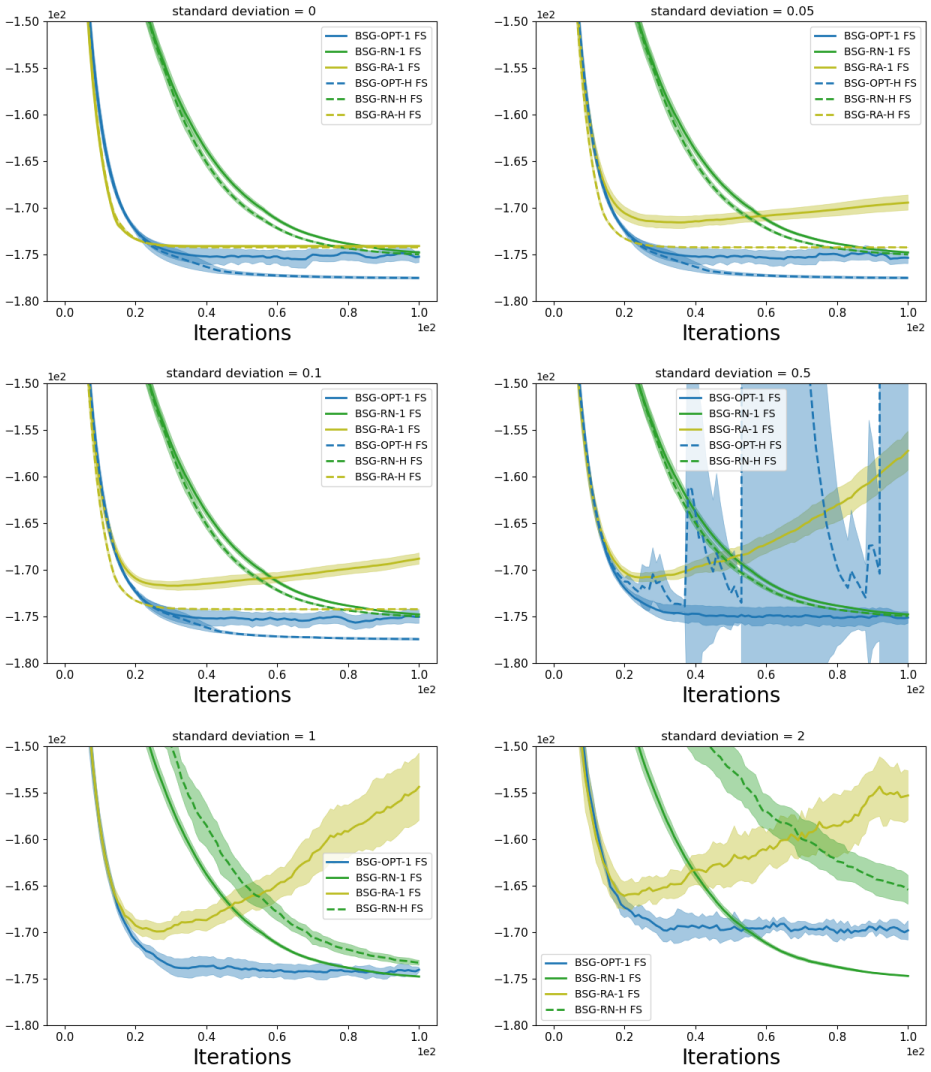

Figure 6: Results for Problem 3 in the stochastic non-separable LL case for different standard deviation values of the stochastic gradient and Hessian estimates (for $\bar{n}$ in Table 1 equal to 50). The vertical axis represents the values of f_{OPT}, f_{RN}, and f_{RA}.

7.1 The case multi-objective single-objective

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

$$\min_{x \in \mathbb{R}^n} \quad F_u(x, y(x)) \text{ s.t. } x \in X, \tag{7.1}$$

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) [10, 15] and stochastic multi-gradients [9, 25]. In (7.1), 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 $f^i(x) = f^i_u(x, y(x))$ as $\nabla f^i(x)$, with $i \in \{1, \ldots, p\}$, the adjoint multi-gradient can be obtained by first solving the QP subproblem

$$\min_{\delta \in \mathbb{R}^p} \quad \left\| \sum_{i=1}^{p} \delta_i \nabla f^i(x) \right\|^2 \qquad \text{s.t.} \qquad \delta \in \Delta, \tag{7.2}$$

where $\Delta = \{\delta \in \mathbb{R}^p : \sum_{i=1}^p \delta_i = 1, \delta_i \geq 0 \ \forall i \in \{1, \ldots, p\}\}$ 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 (3.3), 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 (7.2). In the stochastic case, all the gradients and Hessians in the p adjoint gradients used in problem (7.2) are replaced by their corresponding stochastic estimates.

7.2 The case multi-objective multi-objective

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 3–5. In particular, one can use an adjoint multi-gradient method to solve the UL problem (see Subsection 7.1) 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 (7.2) in the LL single-objective case), for all $i \in \{1, \ldots, p\}$. 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 $f^i_{\text{OPT}}(x, \lambda) = f^i_u(x, y(x, \lambda))$ are given by $\nabla f^i_{\text{OPT}} = (\nabla_x f^i_{\text{OPT}}, \nabla_\lambda f^i_{\text{OPT}})$. In the risk-neutral case, each individual objective function is given by $f^i_{\text{RN}}(x) = (1/N) \sum_{t=1}^N f^i_u(x, y_{\lambda^t}(x))$, and the resulting gradients are $\nabla f^i_{\text{RN}}(x) = (1/N) \sum_{t=1}^N \nabla f^{i\,t}_{\text{RN}}(x)$, where $f^{i\,t}_{\text{RN}}(x) = f^i_u(x, y_{\lambda^t}(x))$ for all $i \in \{1, \ldots, p\}$ and $t \in \{1, \ldots, N\}$. Developing an algorithm for the risk-averse case by following the approach introduced in Section 5 leads to a constrained problem like (5.4) 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.

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] 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.

[3] 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.

[4] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, USA, 2000.

[5] N. Couellan and W. Wang. Bi-level stochastic gradient for large scale support vector machine. *Neurocomputing*, 153:300–308, 2015.

[6] 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.

[7] J.M. Danskin. *The Theory of Max-Min and its Application to Weapons Allocation Problems*. Econometrics and Operations Research. Springer, 1967.

[8] S. Dempe and P. Mehlitz. Semivectorial bilevel programming versus scalar bilevel programming. *Optimization*, 69:657–679, 2020.

[9] J. A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *C. R. Math. Acad. Sci. Paris*, 350:313–318, 2012.

[10] L. G. Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175:395–414, 2005.

[11] M. Ehrgott. *Multicriteria Optimization*, volume 491. Springer Science & Business Media, Berlin, 2005.

[12] G. Eichfelder. Multiobjective bilevel optimization. *Math. Program.*, 123:419–449, jun 2010.

[13] G. Eichfelder. *Methods for multiobjective bilevel optimization*, pages 423–449. Springer International Publishing, Cham, 2020.

[14] A. V. Fiacco. Optimal value differential stability bounds under the mangasarian-fromovitz constraint qualification. *Mathematical Programming with Data Perturbations*, 2:65–90, 1983.

[15] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.*, 51:479–494, 2000.

[16] S. Ghadimi and M. Wang. Approximation methods for bilevel programming. *arXiv e-prints*, art. arXiv:1802.02246, 2018.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] A. M. Lessin, B. J. Lunday, and R. R. Hill. A multi-objective, bilevel sensor relocation problem for border security. *IISE Transactions*, 51:1091–1109, 2019.

[24] 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.

[25] S. Liu and L. N. Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *arXiv e-prints*, art. arXiv:1907.04472, July 2019.

[26] 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.

[27] 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.

[28] 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.

[29] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, second edition, 2006.

[30] M. S. Ozdayi, M. Kantarcioglu, and R. Iyer. BiFair: Training fair models with bilevel optimization. *arXiv e-prints*, art. arXiv:2106.04757, June 2021.

[31] 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.

[32] 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.

[33] A. Sinha and K. Deb. Towards understanding evolutionary bilevel multi-objective optimization algorithm. *IFAC Proceedings Volumes*, 42:338–343, 2009.

[34] 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.

[35] 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.