The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning

S. Liu[*] L. N. Vicente[†]

July 9, 2019

Abstract

Optimization of conflicting functions is of paramount importance in decision making, and real world applications frequently involve data that is uncertain or unknown, resulting in multi-objective optimization (MOO) problems of stochastic type. We study the stochastic multi-gradient (SMG) method, seen as an extension of the classical stochastic gradient method for single-objective optimization.

At each iteration of the SMG method, a stochastic multi-gradient direction is calculated by solving a quadratic subproblem, and it is shown that this direction is biased even when all individual gradient estimators are unbiased. We establish rates to compute a point in the Pareto front, of order similar to what is known for stochastic gradient in both convex and strongly convex cases. The analysis handles the bias in the multi-gradient and the unknown a priori weights of the limiting Pareto point.

The SMG method is framed into a Pareto-front type algorithm for the computation of the entire Pareto front. The Pareto-front SMG algorithm is capable of robustly determining Pareto fronts for a number of synthetic test problems. One can apply it to any stochastic MOO problem arising from supervised machine learning, and we report results for logistic binary classification where multiple objectives correspond to distinct-sources data groups.

Keywords: Multi-Objective Optimization, Pareto Front, Stochastic Gradient Descent, Supervised Machine Learning.

1 Introduction

In multi-objective optimization (MOO) one attempts to simultaneously optimize several, potentially conflicting functions. MOO has wide applications in all industry sectors where decision making is involved due to the natural appearance of conflicting objectives or criteria. Applications span across applied engineering, operations management, finance, economics and social sciences, agriculture, green logistics, and health systems. When the individual objectives are

[*] Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA (sul217@lehigh.edu).

[†] Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA and Centre for Mathematics of the University of Coimbra (CMUC) (lnv@lehigh.edu).
Support for this author was partially provided by FCT/Portugal under grants UID/MAT/00324/2019 and P2020 SAICTPAC/0011/2015.

conflicting, no single solution exists that optimizes all of them simultaneously. The goal of MOO is then to find Pareto optimal solutions (also known as efficient points), roughly speaking points for which no other combination of variables leads to a simultaneous improvement in all objectives. The determination of the set of Pareto optimal solutions helps decision makers to define the best trade-offs among the several competing criteria.

We start by introducing an MOO problem consisting of the simultaneous minimization of m individual functions

$$\begin{array}{ll} \min & H(x) = (h_1(x), \ldots, h_m(x))^\top \\ \text{s.t.} & x \in \mathcal{X}, \end{array} \tag{1}$$

where $h_i : \mathbb{R}^n \to \mathbb{R}$ are real valued functions and $\mathcal{X} \subseteq \mathbb{R}^n$ represents a feasible region. We say that the MOO problem is smooth if all objective functions h_i are continuously differentiable. Assuming that no point may exist that simultaneously minimizes all objectives, the notion of Pareto dominance is introduced to compare any two given feasible points $x, y \in \mathcal{X}$. One says that x dominates y if $H(x) < H(y)$ componentwise. A point $x \in \mathcal{X}$ is a Pareto minimizer if it is not dominated by any other point in $\mathcal{X}$. The set of all Pareto minimizers includes the possible multiple minimizers of each individual function. If we want to exclude such points, one can consider the set of strict Pareto minimizers $\mathcal{P}$, by rather considering a weaker form of dominance (meaning x weakly dominates y if $H(x) \leq H(y)$ componentwise and $H(x) \neq H(y)$)[1]. In this paper we can broadly speak of Pareto minimizers as the first-order optimality condition considered will be necessary for both Pareto optimal sets. An important notion in MOO is the Pareto front $H(\mathcal{P})$, formed by mapping all elements of $\mathcal{P}$ into the decision space $\mathbb{R}^m$, $H(\mathcal{P}) = \{H(x) : x \in \mathcal{P}\}$.

1.1 Deterministic multi-objective optimization

If one calculates the Pareto front, or a significant portion of it, using an *a posteriori* methodology, then decision-making preferences can then be expressed upon the determined Pareto information. This approach contrasts with methods that require an *a priori* input from the decision maker in the decision space, such as a utility function or a ranking of importance of the objectives [21, 32].

A main class of MOO methods apply scalarization, reducing the MOO problem to a single objective one, whose solution is a Pareto minimizer. These methods require an *a priori* selection of the parameters to produce such an effect. The weighted-sum method is simply to assign each objective function $h_i(x)$ a nonnegative weight a_i and minimize the single objective $S(x, a) = \sum_{i=1}^m a_i h_i(x)$ subject to the problem constraints (see, for instance, [27]). If all objective functions are convex, by varying the weights in a simplex set one is guaranteed to determine the entire Pareto front. The ϵ–constraint method [29] consists of minimizing one objective, say $h_i(x)$, subject to additional constraints that $h_j(x) \leq \epsilon_j$ for all $j \neq i$, where $\epsilon_j \geq \min_{x \in \mathcal{X}} h_j(x)$ is an upper bound h_j is allowed to take. Although scalarization methods are conceptually simple, they exhibit some drawbacks: 1) Weights are difficult to preselect, especially when objectives have different magnitudes. Sometimes, the choice of parameters can be problematic, e.g., producing infeasibility in the ϵ–constraint method; 2) In the weighted-sum method, it is frequently observed

[1] Note that both dominance conditions above stated induce a strict partial ordering of the points in $\mathbb{R}^n$. Also, subjacent to such orderings is the cone corresponding to the nonnegative orthant $K = \{v \in \mathbb{R}^n : v_i \geq 0, i = 1, \ldots, n\}$. In fact, x dominates y if and only if $H(y) - H(x) \in \text{int}(K)$, and x weakly dominates y if and only if $H(y) - H(x) \in K \setminus \{0\}$. Broadly speaking any pointed convex cone K will induce in these two ways a strict partial ordering.

(even for convex problems) that an evenly distributed set of weights in a simplex fails to produce an even distribution of Pareto minimizers in the front. 3) It might be impossible to find the entire Pareto front if some of the objectives are nonconvex, as it is the case for the weighted-sum method. There are scalarization methods which have an *a posteriori* flavor like the so-called normal boundary intersection method [13], and which are able to produce a more evenly distributed set of points on the Pareto front given an evenly distributed set of weights (however solutions of the method subproblems may be dominated points in the nonconvex case [26]).

Nonscalarizing *a posteriori* methods attempt to optimize the individual objectives simultaneously in some sense. The methodologies typically consist of iteratively updating a list of nondominated points, with the goal of approximating the Pareto front. To update such iterate lists, some of these *a posteriori* methods borrow ideas from population-based heuristic optimization, including Simulated Annealing, Evolutionary Optimization, and Particle Swarm Optimization. NSGA-II [14] and AMOSA [4] are two well-studied population-based heuristic algorithms designed for MOO. However, no theoretical convergence properties can be derived under reasonable assumptions for these methods, and they are slow in practice due to the lack of first-order principles. Other *a posteriori* methods update the iterate lists by applying steps of rigorous MOO algorithms designed for the computation of a single point in the Pareto front. Such rigorous MOO algorithms have resulted from generalizing classical algorithms of single-objective optimization to MOO.

As mentioned above, a number of rigorous algorithms have been developed for MOO by extending single-objective optimization counterparts. A common feature of these MOO methods is the attempt to move along a direction that simultaneously decreases all objective functions. In most instances it is possible to prove convergence to a first-order stationary Pareto point. Gradient descent is a first example of such a single-objective optimization technique that led to the multi-gradient (or multiple gradient) method for MOO [23] (see also [20, 18, 16, 17]). As analyzed in [24], it turns out that the multi-gradient method proposed by [23] shares the same convergence rates as in the single objective case, for the various cases of nonconvex, convex, and strongly convex assumptions. Other first-order derivative-based methods that were extended to MOO include proximal methods [6], nonlinear conjugate gradient methods [34], and trust-region methods [36, 42]. Newton's method for multi-objective optimization, further using second-order information, was first presented in [22] and later studied in [19]. For a complete survey on multiple gradient-based methods see [26]. Even when derivatives of the objective functions are not available for use, rigorous techniques were extended along the same lines from one to several objectives, an example being the the so-called direct multi-search algorithm [12].

1.2 Stochastic multi-objective optimization

1.2.1 Single objective

Many practical optimization models involve data parameters that are unknown or uncertain, examples being demand or return. In some cases the parameters are confined to sets of uncertainty, leading to robust optimization, where one tries to find a solution optimized against a worst-case scenario. In stochastic optimization/programming, data parameters are considered random variables, and frequently some estimation can be made about their probability distributions. Let us consider the unconstrained optimization of a single function $f(x, w)$ that depends on the decision variables x and on unknown/uncertain parameters w. The goal of stochastic optimization is to seek a solution that optimizes the expectation of f taken with respect to the random parameters

$$\min \; f(x) \; = \; \mathbb{E}[f(x,w)], \tag{2}$$

where $w \in \mathbb{R}^p$ is a random vector defined on a probability space (with probability measure independent from x), for which we assume that i.i.d. samples w can be observed or generated. An example of interest to us is classification in data analysis and learning, where one wants to build a predictor (defined by x) that maps features into labels (the data w) by minimizing some form of misclassification. The objective function $f(x)$ in (2) is then called the expected risk (of misclassification), for which there is no explicit form since pairs of features and labels are drawn according to a unknown distribution.

There are two widely-used approaches for solving problem (2), the sample average approximation (SAA) method and the stochastic approximation (SA) method. Given N i.i.d. samples $\{w^j\}_{j=1}^N$, one optimizes in SAA (see [31, 41]) an empirical approximation of the expected risk

$$\min \; f^N(x) \; = \; \frac{1}{N}\sum_{j=1}^{N} f(x,w^j). \tag{3}$$

The SA method becomes an attractive approach in practice when the explicit form of the gradient $\nabla f(x)$ for (2) is not accessible or the gradient $\nabla f^N(x)$ for (3) is too expensive to compute when N is large. The earliest prototypical SA algorithm, also known as stochastic gradient (SG) algorithm, dates back to the paper [38]; and the classical convergence analysis goes back to the works [11, 39]. In the context of solving (2), the SG algorithm is defined by $x_{k+1} = x_k - \alpha_k \nabla f(x_k, w_k)$, where w_k is a copy of w and α_k is a positive stepsize. When solving problem (3), w_k may just be a random sample uniformly taken from $\{w^1, \ldots, w^N\}$. Computing the stochastic gradient $-\nabla f(x_k, w_k)$ based on a single sample makes each iterate of the SG algorithm very cheap. However, note that only the expectation of $-\nabla f(x_k, w_k)$ is descent for f at x_k, and therefore the performance of the SG algorithm is quite sensitive to the variance of the stochastic gradient. A well-known idea to improve its performance is the use of a batch gradient at each iterate, namely updating each iterate by $x_{k+1} = x_k - \frac{\alpha_k}{|S_k|}\sum_{j\in S_k} \nabla f(x_k, w^j)$, where S_k is a minibatch sample from $\{1, \ldots, N\}$ of size $|S_k|$. More advanced variance reduction techniques can be found in [15, 30, 35, 40] (see the review [7]).

1.2.2 Multiple objectives

When the individual objectives have the form in (2), we face a stochastic multi-objective optimization (SMOO) problem:

$$\begin{array}{ll} \min & F(x) = (f_1(x), \ldots, f_m(x))^\top = (\mathbb{E}[f_1(x,w)], \ldots, \mathbb{E}[f_m(x,w)])^\top \\ \text{s.t.} & x \in \mathcal{X}, \end{array} \tag{4}$$

where $f_i(x)$ denotes now the i-th objective function value and $f_i(x,w)$ is the i-th stochastic function value with respect to the random parameters $w \in \mathbb{R}^{m\times p}$. In the finite sum case (3) one has that $\mathbb{E}[f_i(x,w)]$ is equal to or can be approximated by

$$f_i^N(x) \; = \; \frac{1}{N}\sum_{j=1}^{N} f_i(x,w^j). \tag{5}$$

Our work assumes that $\mathcal{X}$ does not involve uncertainty (see the survey [2]) for problems with both stochastic objectives and constraints).

The main approaches for solving the SMOO problems are classified into two categories [1]: the *multi-objective* methods and the *stochastic* methods. The multi-objective methods first reduce the SMOO problem into a deterministic MOO problem, and then solve it by techniques for deterministic MOO (see Subsection 1.1). The stochastic methods first aggregate the SMOO problem into a single objective stochastic problem and then apply single objective stochastic optimization methods (see Subsection 1.2.1). Both approaches have disadvantages [8]. Note that the stochastic objective functions $f_i, i = 1, \ldots, m$, may be correlated to each other as they involve a common random variable $w \in \mathbb{R}^{m \times p}$. Without taking this possibility into consideration, the multi-objective methods might simplify the problem by converting each stochastic objective to a deterministic counterpart independently of each other. As for the stochastic methods, they obviously inherit the drawbacks of *a priori* scalarizarion methods for deterministic MOO. We will nevertheless restrict our attention to multi-objective methods by assuming that the random variables in the individual objectives are independent of each other.

1.3 Contributions of this paper

This paper contributes to the solution of stochastic multi-objective optimization (SMOO) problems of the form (4) by providing a understanding of the behavior and basic properties of the stochastic multi-gradient (SMG) method, also called stochastic multiple gradient method [37]. Stochastic gradient descent is well studied and understood for single-objective optimization. Deterministic multi-gradient descent is also well understood for MOO. However, little is known yet about stochastic multi-gradient descent for stochastic MOO. The authors in [37] introduced and analyzed it, but failed to identify a critical property about the stochastic multi-gradient direction, and as a consequence analyzed the method under strong and unnatural assumptions. Moreover, they did not present its extension from an algorithm that produces a single point in the Pareto front to one that computes the entire Pareto front.

The steepest descent direction for deterministic MOO results from the solution of a subproblem where one tries to compute a direction that is the steepest among all functions, subject to some form of Euclidean regularization. The dual of this problem shows that this is the same as calculating the negative of the minimum-norm convex linear combination of all the individual gradients (and the coefficients of such a linear convex combination form a set of simplex weights). From here it becomes then straightforward how to compute a direction to be used in the stochastic multi-gradient method, by simply feeding into such a subproblem unbiased estimations of the individual gradients (the corresponding weights are an approximation or estimation of the true ones). However it turns out the Euclidean regularization or minimum-norm effect introduces a bias in the overall estimation. A practical implementation and a theoretical analysis of the method have necessarily to take into account the biasedness of the stochastic multi-gradient.

In this paper we first study the bias of the stochastic multi-gradient direction and derive a condition for the amount of biasedness that is tolerated to achieve convergence at the appropriate rates. Such a condition will depend on the stepsize but can be enforced by increasing the batch size used to estimate the individual gradients. Another aspect that introduces more complexity in the MOO case is not knowing the limiting behavior of the approximate weights generated by the algorithm when using sampled gradients, or even of the true weights if the subproblem would be solved using the true gradients. In other words, this amounts to say that we do not know which point in the Pareto front the algorithm is targeting at. We thus develop a convergence analysis measuring the expected gap between $S(x_k, \lambda_k)$ and $S(x_*, a_k)$, for various possible selections of a_k as approximations for λ_*, where x_k is the current iterate, λ_k are the true weighs, and x_* is a Pareto optimal solution (with corresponding weights λ_*). The choice $a_k = \lambda_*$ requires however a stronger assumption, essentially saying that λ_k identifies well the optimal role of λ_*. Our convergence analysis shows that the stochastic multi-gradient algorithm exhibits convergence rates similar as in the single stochastic gradient method, i.e., $\mathcal{O}(1/k)$ for strongly convexity and $\mathcal{O}(1/\sqrt{k})$ for convexity.

The practical solution of many MOO problems requires however the calculation of the entire Pareto front. Having such a goal in mind also for the stochastic MOO case, we propose a Pareto-front multi-gradient stochastic (PF-SMG) method that iteratively updates a list of non-dominated points by applying a certain number of steps of the stochastic multi-gradient method at each point of the list. Such process generates a number of points which are then added to the list. The main iteration is ended by removing possible dominated points from the list. We tested our Pareto-front stochastic multi-gradient method, using synthesis MOO problems [12] to which noise was artificially added, and then measured the quality of the approximated Pareto fronts in terms of the so-called Purity and Spread metrics. The new algorithm shows satisfactory performance when compared with a corresponding deterministic counterpart.

We have applied the Pareto-Front SMG algorithm to stochastic MOO problems arising from supervised machine learning, in the setting of logistic binary classification where multiple objectives correspond to different sources of data within a set. The determination of the Pareto front can help identifying classifiers that trade-off such sources or contexts, thus improving the *fairness* of the classification process.

1.4 Organization of this paper

In the context of deterministic MOO, we review in Section 2 the first-order necessary condition for Pareto optimality and the subproblems for computing the common descent direction, denoted here by multi-gradient direction. Section 3 introduces the Stochastic Multi-Gradient (SMG) algorithm in detail, and Section 4 reports on the existence of biasedness in the stochastic multi-gradients used in the algorithm. The convergence rates for both convex and strongly convex cases are derived in Section 5. The Pareto-Front Stochastic Multi-Gradient algorithm (PF-SMG) is outlined in Section 6. Our numerical experiments for synthetic and learning problems are reported in Section 7, and the paper in concluded in Section 8 with final remarks and prospects of future work.

2 Pareto stationarity and common descent direction in the deterministic multi-objective case

The simplest descent method for solving smooth unconstrained MOO problems, i.e. problem (1) with $\mathcal{X} = \mathbb{R}^n$, is the multi-gradient method proposed originally in [23] and further developed in [16, 20]. Each iterate takes a step of the form $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a positive stepsize and d_k is a common descent direction at the current iteration x_k.

A necessary condition for a point x_k to be a (strict or nonstrict) Pareto minimizer of (1) is that there does not exist any direction that is first-order descent for all the individual objectives, i.e.,

$$\text{range}\,(\nabla J_H(x_k)) \cap (-\mathbb{R}^m_{++}) \;=\; \emptyset, \tag{6}$$

where $\mathbb{R}^m_{++}$ is the positive orthant cone and $\nabla J_H(x_k)$ denotes the Jacobian matrix of H at x_k. Condition (6) characterizes first-order Pareto stationary. In fact, at such a nonstationary point x_k, there must exist a descent direction $d \in \mathbb{R}^n$ such that $\nabla h_i(x_k)^\top d < 0$, $i = 1, \ldots, m$, and one could decrease all functions along d.

When $m = 1$ we simply take $d_k = -\nabla h_1(x_k)$ as the steepest descent or negative gradient direction, and this amounts to minimize $-\nabla h_1(x_k)^\top d + (1/2)\|d\|^2$ in d. In MOO ($m > 1$), the steepest common descent direction [23] is defined by minimizing the amount of first-order Pareto stationarity, also in a regularized Euclidean sense,

$$\begin{array}{rcl} (d_k, \beta_k) \;\in\; \text{argmin}_{d\in\mathbb{R}^n,\beta\in\mathbb{R}} & & \beta + \dfrac{1}{2}\|d\|^2 \\ & \text{s.t.} & \nabla h_i(x_k)^\top d - \beta \leq 0, \; \forall i = 1, \ldots, m. \end{array} \tag{7}$$

If x_k is first-order Pareto stationary, then $(d_k, \beta_k) = (0, 0) \in \mathbb{R}^{n+1}$, and if not, $\nabla h_i(x_k)^\top d_k \leq \beta_k < 0$, for all $i = 1, \ldots, m$ (see [23]). The direction d_k minimizes $\max_{1\leq i\leq m}\{-\nabla h_i(x_k)^\top d_k\} + (1/2)\|d\|^2$.

It turns out that the dual of (7) is the following subproblem

$$\begin{array}{rcl} \lambda^k \;\in\; \text{argmin}_{\lambda\in\mathbb{R}^m} & & \left\| \displaystyle\sum_{i=1}^m \lambda_i \nabla h_i(x_k) \right\|^2 \\ & \text{s.t.} & \lambda \in \Delta^m, \end{array} \tag{8}$$

where $\Delta^m = \{\lambda : \sum_{i=1}^m \lambda_i = 1, \lambda_i \geq 0, \forall i = 1, ..., m\}$ denotes the simplex set. Subproblem (8) reflects the fact that the common descent direction is pointing opposite to the minimum-norm vector in the convex hull of the gradients $\nabla h_i(x_k), i = 1, \ldots, m$. Hence, the common descent direction, called in this paper a negative multi-gradient, is written as $d_k = -\sum_{i=1}^m \lambda_i^k \nabla h_i(x_k)$. In the single objective case ($m = 1$), one recovers $d_k = -\nabla h_1(x_k)$. If x_k is first-order Pareto stationary, then the convex hull of the individual gradients contains the origin, i.e.,

$$\exists \lambda \;\in\; \Delta^m \text{ such that } \sum_{i=1}^m \lambda_i \nabla h_i(x_k) \;=\; 0. \tag{9}$$

When all the objective functions are convex, we have $x_k \in \mathcal{P}$ if and only if x_k is Pareto first-order stationary [28, 32].

The multi-gradient algorithm [23] consists of taking $x_{k+1} = x_k + \alpha_k d_k$, where d_k results from the solution of any of the above subproblems and α_k is a positive stepsize. The norm of d_k is a natural stopping criterion. Selecting α_k either by backtracking until an appropriate sufficient decrease condition is satisfied or by taking a fixed stepsize inversely proportional to the maximum of the Lipschitz constants of the gradients of the individual gradients leads to the classical sublinear rates of $1/\sqrt{k}$ and $1/k$ in the nonconvex and convex cases, respectively, and to a linear rate in the strongly convex case [24].

3 The stochastic multi-gradient method

Let us now introduce the stochastic multi-gradient (SMG) algorithm for the solution of the stochastic MOO problem (4). For this purpose let $\{w_k\}_{k \in \mathbb{N}}$ be a sequence of copies of the random variable w. At each iteration we sample stochastic gradients $g_i(x_k, w_k)$ as approximations of the true gradients $\nabla f_i(x_k)$, $i = 1 \ldots, m$. The stochastic multi-gradient is then computed by replacing the true gradients $\nabla f_i(x_k)$ in subproblem (8) by the corresponding stochastic gradients $g_i(x_k, w_k)$, leading to the following subproblem:

$$\begin{aligned} \lambda^g(x_k, w_k) \; \in \; \operatorname{argmin}_{\lambda \in \mathbb{R}^m} \quad & \left\| \sum_{i=1}^m \lambda_i g_i(x_k, w_k) \right\|^2 \\ \text{s.t.} \quad & \lambda \in \Delta^m, \end{aligned} \tag{10}$$

where the convex combination coefficients $\lambda_k^g = \lambda^g(x_k, w_k)$ depend on x_k and on the random variable w_k. Let us denote the stochastic multi-gradient by

$$g(x_k, w_k) \;=\; \sum_{i=1}^m \lambda_i^g(x_k, w_k) g_i(x_k, w_k). \tag{11}$$

Analogously to the unconstrained deterministic case, each iterative update of the SMG algorithm takes the form $x_{k+1} = x_k - \alpha_k g(x_k, w_k)$, where α_k is a positive step size and $g(x_k, w_k)$ is the stochastic multi-gradient. More generally, when considering a closed and convex constrained set $\mathcal{X}$ different from $\mathbb{R}^n$, we need to first orthogonally project $x_k - \alpha_k g(x_k, w_k)$ onto $\mathcal{X}$ (such projection is well defined and results from the solution of a convex optimization problem). The SMG algorithm is described as follows.

Algorithm 1 Stochastic Multi-Gradient (SMG) Algorithm

```
1: Choose an initial point $x_0 \in \mathbb{R}^n$ and a step size sequence $\{\alpha_k\}_{k \in \mathbb{N}} > 0$.
2: for $k = 0, 1, \ldots$ do
3:     Compute the stochastic gradients $g_i(x_k, w_k)$ for the individual functions, $i = 1, \ldots, m$.
4:     Solve problem (10) to obtain the stochastic multi-gradient (11) with $\lambda_k^g = \lambda^g(x_k, w_k) \in \Delta^m$.
5:     Update the next iterate $x_{k+1} = P_{\mathcal{X}}(x_k - \alpha_k g(x_k, w_k))$.
6: end for
```

As in the stochastic gradient method, there is also no good stopping criterion for the SMG algorithm, and one may have just to impose a maximum number of iterations.

4 Biasedness of the stochastic multi-gradient

Figure 1 provides us the intuition for Subproblems (8) and (10) and their solutions when $n = m = 2$. In this section for simplicity we will omit the index k. Let g_1^1 and g_2^1 be two unbiased estimates of the true gradient $\nabla f_1(x)$ for the first objective function, and g_1^2 and g_2^2 be two unbiased estimates of the true gradient $\nabla f_2(x)$ for the second objective function. Then, g_1

and g_2, the stochastic multi-gradients from solving (10), are estimates of the true multi-gradient g obtained from solving (8).

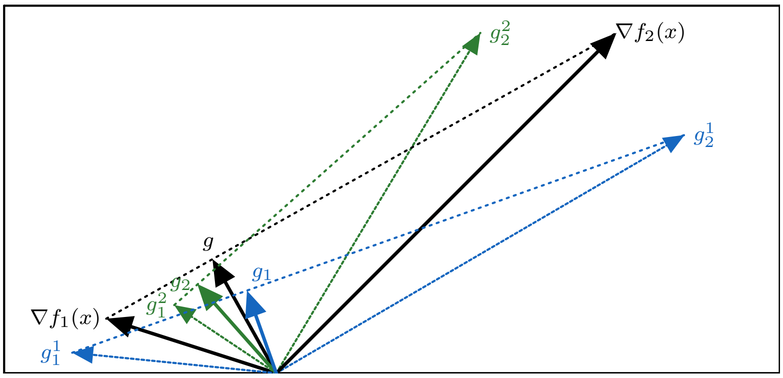
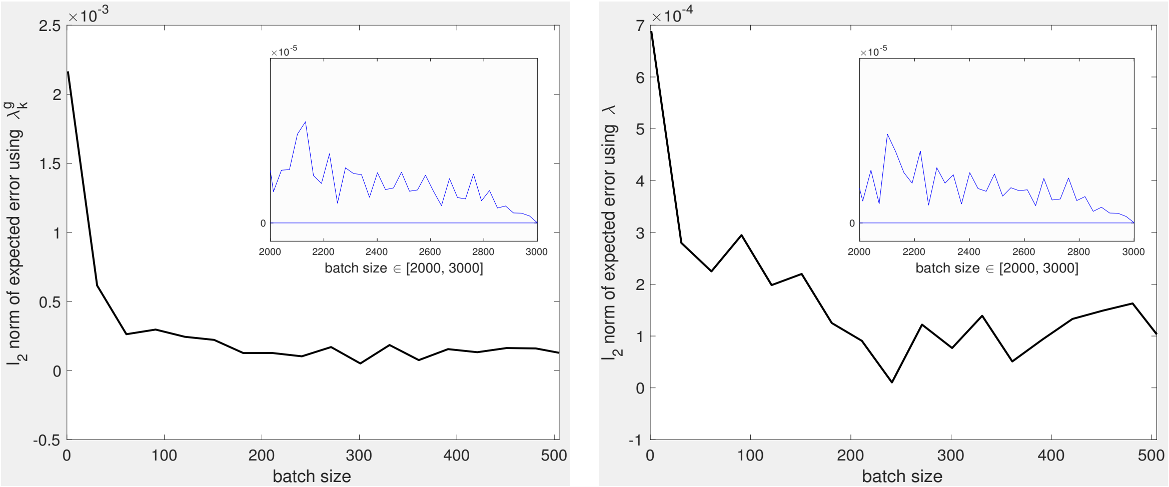


Figure 1: Illustration of the solutions of Subproblems (8) and (10).

As we mention in the introduction, let $S(x, \lambda) = \sum_{i=1}^{m} \lambda_i f_i(x)$ denote the weighted true function and $\nabla_x S(x, \lambda) = \sum_{i=1}^{m} \lambda_i \nabla f_i(x)$ the corresponding gradient.

When $m = 1$, recall that is classical to assume that the stochastic gradients are unbiased estimates of the corresponding true gradients. In the MOO case ($m > 1$), even if $g_i(x, w)$ are unbiased estimates of $\nabla f_i(x)$ for all $i = 1, \ldots, m$, the stochastic multi-gradient $g(x, w)$ resulting from solving (10) is a biased estimate of $\mathbb{E}_w[\nabla_x S(x, \lambda^g)]$, where λ^g are the convex combination coefficients associated with $g(x, w)$, or even of the true multi-gradient $\nabla_x S(x, \lambda)$, where λ are now the coefficients that result from solving (8) with true gradients $\nabla f_i(x)$, $i = 1, \ldots, m$. Basically, the solution of the QP (10) acts as a mapping on the unbiased stochastic gradients $g_i(x, w)$, $i = 1, \ldots, m$, introducing biasedness in the mapped outcome (the stochastic multi-gradient $g(x, w)$).

Let us observe the amount of biasedness by looking at the norm of the expected error $\|\mathbb{E}_w[g(x, w) - \nabla_x S(x, \lambda^g)]\|$ in an experiment with $n = 4$ and $m = 2$, where each objective function is a finite sum of the form (5). The true gradients of the two objectives were first randomly generated with norms 37.18 and 40.64 respectively. For each objective, we then generated 3000 four-dimensional stochastic gradients from a normal distribution with mean 0 and variance 0.2, that will form the gradients of the $N = 3000$ terms in (5). A batch size specifies for each objective how many samples in a batch are uniformly drawn from the the set of the 3000 stochastic gradients. (For each batch size, we drew 10000 batches and took means.) Figure 2 (a) confirms the existence of biasedness in the stochastic multi-gradient $g(x, w)$ as an approximation to $\mathbb{E}_w[\nabla_x S(x, \lambda^g)]$. It is observed that the biasedness does indeed decrease as the batch size increases, and that it eventually vanishes in the full batch regime (where Subproblems (8) and (10) become identical).



(a) Using λ^g.

(b) Using λ.

Figure 2: Biasedness decreases as the batch size increases: $m = 2$, $n = 4$, and $N = 3000$.

Biasedness is also present when we look at the norm of the expected error $\|\mathbb{E}_w[g(x,w)] - \nabla_x S(x,\lambda)\|$, using the true coefficients λ. In the same setting of the previous experiment, Figure 2 (b) shows that biasedness still exists, although in a smaller quantity than when using λ^g.

5 Convergence rates for the stochastic multi-gradient method

In this section, the convergence theory of the simple stochastic gradient method is extended to stochastic MOO. We prove sublinear convergence rates of the order of $1/k$ and $1/\sqrt{k}$ for strongly convex and convex cases, respectively, when approaching a point in the Pareto front. Let us start by formulating the assumptions that are common to both cases. First of all, as in the case $m = 1$, we assume that all the objective functions in problem (4) are sufficiently smooth.

Assumption 5.1 ***(Lipschitz continuous gradients)*** *All objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable with gradients ∇f_i Lipschitz continuous with Lipschitz constants $L_i > 0$, $i = 1, \ldots, m$, i.e., $\|\nabla f_i(x) - \nabla f_i(\bar{x})\| \leq L_i \|x - \bar{x}\|, \forall (x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n$.*

Assumption 5.1 implies smoothness of the weighted true function $S(x,\lambda) = \sum_{i=1}^m \lambda_i f_i(x)$. In fact, given any $\lambda \in \Delta^m$, the weighted true function $S(x,\lambda)$ has Lipschitz continuous gradients in x with constant $L = \max_{1 \leq i \leq m}\{L_i\}$, i.e.,

$$\|\nabla_x S(x,\lambda) - \nabla_x S(\bar{x},\lambda)\| \;\leq\; L\|x - \bar{x}\|, \quad \forall (x,\bar{x}) \;\in\; \mathbb{R}^n \times \mathbb{R}^n. \tag{12}$$

We will use $\mathbb{E}_{w_k}[\cdot]$ to denote the expected value taken with respect to w_k. Notice that x_{k+1} is a random variable depending on w_k whereas x_k does not.

Now we propose our assumptions on the amount of biasedness and variance of the stochastic multi-gradient $g(x_k, w_k)$. As commonly seen in the literature of the standard stochastic gradient method, we assume that the individual stochastic gradients $g_i(x_k, w_k), i = 1, \ldots, m$, are unbiased estimates of the corresponding true gradients and that their variance is bounded by the size of these gradients (Assumptions (a) and (c) below). However, an assumption is also needed to bound the amount of biasedness of the stochastic multi-gradient in terms of the stepsize α_k (Assumption (b) below).

Assumption 5.2 *For all objective functions f_i, $i = 1, \ldots, m$, and iterates $k \in \mathbb{N}$, the individual stochastic gradients $g_i(x_k, w_k)$ satisfy the following:*

(a) ***(Unbiasedness)*** $\mathbb{E}_{w_k}[g_i(x_k, w_k)] \; = \; \nabla f_i(x_k)$.

(b) ***(Bound on the first moment)*** *There exist positive scalars $C_i > 0$ and $\hat{C}_i > 0$ such that*

$$\mathbb{E}_{w_k}\left[\|g_i(x_k, w_k) - \nabla f_i(x_k)\|\right] \; \leq \; \alpha_k \left(C_i + \hat{C}_i \|\nabla f_i(x_k)\| \right). \tag{13}$$

(c) ***(Bound on the second moment)*** *There exist positive scalars $G_i > 0$ and $\hat{G}_i > 0$ such that*

$$\mathbb{V}_{w_k}[g_i(x_k, w_k)] \; \leq \; G_i^2 + \hat{G}_i^2 \|\nabla f_i(x_k)\|^2.$$

In fact, based on inequality (13), one can derive an upper bound for the biasedness of the stochastic multi-gradient

$$\begin{aligned}
\left\|\mathbb{E}_{w_k}[g(x_k, w_k) - \nabla_x S(x_k, \lambda_k^g)]\right\| &\leq \mathbb{E}_{w_k}\left[\left\|g(x_k, w_k) - \nabla_x S(x_k, \lambda_k^g)\right\|\right] \\
&= \mathbb{E}_{w_k}\left[\left\|\sum_{i=1}^m (\lambda_k^g)_i (g_i(x_k, w_k) - \nabla f_i(x_k))\right\|\right] \\
&\leq \sum_{i=1}^m \mathbb{E}_{w_k}[\|g_i(x_k, w_k) - \nabla f_i(x_k)\|] \\
&\leq \alpha_k \left(\sum_{i=1}^m C_i + \sum_{i=1}^m \hat{C}_i \|\nabla f_i(x_k)\| \right),
\end{aligned}$$

where the first inequality results from Jensen's inequality in the context of probability theory. As a consequence, we have

$$\|\mathbb{E}_{w_k}[g(x_k, w_k) - \nabla_x S(x_k, \lambda_k^g)]\| \; \leq \; \alpha_k \left(M_1 + M_F \sum_{i=1}^m \|\nabla f_i(x_k)\| \right) \tag{14}$$

with $M_1 = \sum_{i=1}^m C_i$ and $M_F = \max_{1 \leq i \leq m} \hat{C}_i$. Note that we could have imposed directly the assumption

$$\|\mathbb{E}_{w_k}[g(x_k, w_k) - \nabla_x S(x_k, \lambda_k^g)]\| \; \leq \; \alpha_k \left(M_1 + M_F \left\|\mathbb{E}_{w_k}[\nabla_x S(x_k, \lambda_k^g)]\right\| \right),$$

from which then (14) would have easily followed. However we will see later that we will also need the more general version stated in Assumption 5.2 (b).

Using Assumptions 5.2 (a) and (c), we can generalize the bound on the variance of the individual stochastic gradients $g_i(x_k, w_k)$ to the stochastic multi-gradient $g(x_k, w_k)$. In fact we first note that

$$\begin{aligned}
\mathbb{E}_{w_k}[\|g_i(x_k, w_k)\|^2] &= \mathbb{V}_{w_k}[g_i(x_k, w_k)] + \left\|\mathbb{E}_{w_k}[g_i(x_k, w_k)]\right\|^2 \\
&\leq G_i^2 + (\hat{G}_i^2 + 1)\|\nabla f_i(x_k)\|^2,
\end{aligned}$$

from which we then obtain

$$\begin{aligned}\mathbb{E}_{w_k}[\|g(x_k,w_k)\|^2] &= \mathbb{E}_{w_k}\left[\left\|\sum_{i=1}^m(\lambda_k^g)_i g_i(x_k,w_k)\right\|^2\right] \\ &\le \mathbb{E}_{w_k}\left[m\sum_{i=1}^m\|g_i(x_k,w_k)\|^2\right] \\ &\le m\sum_{i=1}^m\left(G_i^2+(\hat{G}_i^2+1)\|\nabla f_i(x_k)\|^2\right) \\ &= G^2+G_V^2\sum_{i=1}^m\|\nabla f_i(x_k)\|^2\end{aligned}$$

with $G^2 = m\sum_{i=1}^m G_i^2$ and $G_V^2 = m\max_{1\le i\le m}(\hat{G}_i^2+1)$. Note that the obtained inequality is consistent with imposing directly a bound of the form

$$\mathbb{E}_{w_k}\left[\|g(x_k,w_k)\|^2\right] \;\le\; G^2+G_V^2\left\|\mathbb{E}_{w_k}[\nabla_x S(x_k,\lambda_k^g)]\right\|^2$$

from the fact that $\|\mathbb{E}_{w_k}[\nabla_x S(x_k,\lambda_k^g)]\|^2 \le m\sum_{i=1}^m\|\nabla f_i(x_k)\|^2$.

We will need the iterates to lie in a bounded set, one of the reasons being the need to bound the norm of the true gradients. We can achieve this by asking $\mathcal{X}$ to be a bounded set (in addition to being closed and convex).

Assumption 5.3 *The feasible region $\mathcal{X}\subset\mathbb{R}^n$ is a bounded set.*

The above assumption implies the existence of an upper bound on the diameter of the feasible region, i.e., there exists a positive constant Θ such that

$$\max_{x,y\in\mathcal{X}}\|x-y\| \;\le\; \Theta \;<\; \infty. \tag{15}$$

Note that from Assumption 5.1 and (15), the norm of the true gradient of each objective function is bounded, i.e., $\|\nabla f_i(x)\|\le M_\nabla + L\Theta$, for $i=1,\ldots,m$, and any $x\in\mathcal{X}$, where M_∇ denotes the largest of the norms of the ∇f_i at an arbitrary point of $\mathcal{X}$. For conciseness, denote $L_{\nabla S} = M_1 + mM_F(M_\nabla+L\Theta)$ and $L_g^2 = G^2 + mG_V^2(M_\nabla+L\Theta)^2$. Hence, we have

$$\left\|\mathbb{E}_{w_k}\left[g(x_k,w_k)-\nabla_x S(x_k,\lambda_k^g)\right]\right\| \;\le\; \alpha_k L_{\nabla S} \tag{16}$$

and

$$\mathbb{E}_{w_k}\left[\|g(x_k,w_k)\|^2\right] \;\le\; L_g^2. \tag{17}$$

Lastly, we need to bound the sensitivity of the solution of the Subproblem (8), a result that follows locally from classical sensitivity theory but that we assume globally too.

Assumption 5.4 ***(Subproblem Lipschitz continuity)*** *The optimal solution of Subproblem* (8) *is a Lipschitz continuous function of the parameters $\{\nabla f_i(x), 1\le i\le m\}$, i.e., there exists a scalar $\beta>0$ such that*

$$\|\lambda^k-\lambda^s\| \;\le\; \beta\left\|\left[(\nabla f_1(x_k)-\nabla f_1(x_s))^\top,\ldots,(\nabla f_m(x_k)-\nabla f_m(x_s))^\top\right]\right\|.$$

As a consequence of the above assumption, the optimal solutions of Subproblems (8) and (10) satisfy

$$\begin{aligned}
\mathbb{E}_{w_k}[\|\lambda_k^g - \lambda^k\|] &\leq \beta \mathbb{E}_{w_k}\left[\left\|[(g_1(x_k,w_k) - \nabla f_1(x_k))^\top, \ldots, (g_m(x_k,w_k) - \nabla f_m(x_k))^\top]\right\|\right] \\
&\leq \beta \sum_{i=1}^{m} \mathbb{E}_{w_k}\left[\|g_i(x_k,w_k) - \nabla f_i(x_k)\|\right] \qquad (18) \\
&\leq \alpha_k(\beta L_{\nabla S}),
\end{aligned}$$

where $L_{\nabla S}$ is the constant defined in (16). Since $\nabla_x S(x,\lambda)$ is a linear function of λ,

$$\left\|\nabla_x S(x_k,\lambda_k^g) - \nabla_x S(x_k,\lambda_k)\right\| \;\leq\; M_S \left\|\lambda_k^g - \lambda_k\right\|,$$

with $M_S = \sqrt{mn}(M_\nabla + L\Theta)$. By taking expectation over w_k and using (18), one obtains

$$\mathbb{E}_{w_k}\left[\left\|\nabla_x S(x_k,\lambda_k^g) - \nabla_x S(x_k,\lambda_k)\right\|\right] \;\leq\; \alpha_k(\beta L_{\nabla S} M_S). \qquad (19)$$

5.1 The strongly convex case

Strongly convexity is the most widely studied setting in stochastic gradient methods. In the context of MOO we impose it in all individual functions.

Assumption 5.5 *(Strong convexity) All objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are strongly convex, i.e., for all $i = 1, \ldots, m$, there exists a scalar $c_i > 0$ such that*

$$f_i(\bar{x}) \;\geq\; f_i(x) + \nabla f_i(x)^\top(\bar{x} - x) + \frac{c_i}{2}\|\bar{x} - x\|^2, \quad \forall (x,\bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Under this assumption all the individual functions have a unique minimizer in $\mathcal{X}$ that is also a Pareto minimizer. We also conclude that the weighted function $S(x,\lambda)$ is strongly convex with constant $c = \min_{1\leq i\leq m}\{c_i\}$, i.e.,

$$S(\bar{x},\lambda) \;\geq\; S(x,\lambda) + \nabla_x S(x,\lambda)^\top(\bar{x} - x) + \frac{c}{2}\|\bar{x} - x\|^2, \quad \forall (x,\bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n. \qquad (20)$$

We are finally ready to prove a convergence rate under strong convexity, showing that a certain weighted function has the potential to decay sublinearly at the rate of $1/k$, as it happens in the stochastic gradient method for $m = 1$. We will use $\mathbb{E}[\cdot]$ to denote the expected value taken with respect to the joint distribution of $\{w_k, k \in \mathbb{N}\}$. The stepsize choice α_k is of diminishing type, in other words it obeys

$$\sum_{k=1}^{\infty} \alpha_k \;=\; \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha_k^2 \;<\; \infty.$$

Theorem 5.1 *(sublinear convergence rate under strong convexity) Let Assumptions 5.1–5.5 hold and x_* be any point in $\mathcal{X}$. Consider a diminishing step size sequence $\alpha_k = \frac{2}{c(k+1)}$. The sequence of iterates generated by Algorithm 1 satisfies*

$$\min_{s=1,\ldots,k} \mathbb{E}[S(x_s,\lambda_s)] - \mathbb{E}[S(x_*,\bar{\lambda}_k)] \;\leq\; \frac{2L_g^2 + 4\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)}{c(k+1)},$$

where $\bar{\lambda}_k = \sum_{s=1}^{k} \frac{s}{\sum_{s=1}^{k} s}\lambda_s \in \Delta^m$.

Proof. For any $k \in \mathbb{N}$, considering that the projection operation is non-expansive, one can write

$$\begin{aligned}\mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] &= \mathbb{E}_{w_k}[\|P_{\mathcal{X}}(x_k - \alpha_k g(x_k, w_k)) - x_*\|^2] \\ &\leq \mathbb{E}_{w_k}[\|x_k - \alpha_k g(x_k, w_k) - x_*\|^2] \\ &= \|x_k - x_*\|^2 + \alpha_k^2 \mathbb{E}_{w_k}[\|g(x_k, w_k)\|^2] \\ &\quad - 2\alpha_k \mathbb{E}_{w_k}[g(x_k, w_k)]^\top (x_k - x_*).\end{aligned} \tag{21}$$

Adding the null term $2\alpha_k(\mathbb{E}_{w_k}[\nabla_x S(x_k, \lambda_k^g)] - \mathbb{E}_{w_k}[\nabla_x S(x_k, \lambda_k^g)] + \nabla_x S(x_k, \lambda_k) - \nabla_x S(x_k, \lambda_k))$ to the right-hand side yields

$$\begin{aligned}\mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] &\leq \|x_k - x_*\|^2 + \alpha_k^2 \mathbb{E}_{w_k}[\|g(x_k, w_k)\|^2] - 2\alpha_k \nabla_x S(x_k, \lambda_k)^\top (x_k - x_*) \\ &\quad + 2\alpha_k \|\mathbb{E}_{w_k}[g(x_k, w_k) - \nabla_x S(x_k, \lambda_k^g)]\| \|x_k - x_*\| \\ &\quad + 2\alpha_k \|\mathbb{E}_{w_k}[\nabla_x S(x_k, \lambda_k^g) - \nabla_x S(x_k, \lambda_k)]\| \|x_k - x_*\|.\end{aligned} \tag{22}$$

Choosing $\lambda = \lambda_k$, $x = x_k$, and $\bar{x} = x_*$ in inequality (20), one has

$$\nabla_x S(x_k, \lambda_k)^\top (x_k - x_*) \;\geq\; S(x_k, \lambda_k) - S(x_*, \lambda_k) + \frac{c}{2}\|x_k - x_*\|^2. \tag{23}$$

Then, plugging inequalities (16), (17), (19), and (23) into (22), we obtain

$$\begin{aligned}\mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] \;\leq\; &(1 - \alpha_k c)\|x_k - x_*\|^2 + \alpha_k^2(L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)) \\ &- 2\alpha_k \mathbb{E}_{w_k}[S(x_k, \lambda^k) - S(x_*, \lambda^k)].\end{aligned}$$

For simplicity denote $M = L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)$. Using $\alpha_k = \frac{2}{c(k+1)}$, and rearranging the last inequality,

$$\begin{aligned}\mathbb{E}_{w_k}[S(x_k, \lambda_k) - S(x_*, \lambda_k)] \;&\leq\; \frac{(1 - \alpha_k c)\|x_k - x_*\|^2 + \alpha_k^2 M - \mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2]}{2\alpha_k} \\ &\leq \frac{c(k-1)}{4}\|x_k - x_*\|^2 - \frac{c(k+1)}{4}\mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] + \frac{M}{c(k+1)}.\end{aligned}$$

Now we replace k by s in the above inequality. Taking the total expectation, multiplying by s on both sides, and summing over $s = 1, \ldots, k$ yields

$$\begin{aligned}\sum_{s=1}^{k} s(\mathbb{E}[S(x_s, \lambda_s)] - \mathbb{E}[S(x_*, \lambda_s)]) \;&\leq\; \sum_{s=1}^{k}\left(\frac{cs(s-1)}{4}\mathbb{E}[\|x_s - x_*\|^2] - \frac{cs(s+1)}{4}\mathbb{E}[\|x_{s+1} - x_*\|^2]\right) \\ &\quad + \sum_{s=1}^{k} \frac{s}{c(s+1)} M \\ &\leq -\frac{c}{4}k(k+1)\mathbb{E}[\|x_{k+1} - x_*\|^2] + \sum_{s=1}^{k} \frac{s}{c(s+1)} M \\ &\leq \frac{k}{c} M.\end{aligned}$$

Dividing both sides of the last inequality by $\sum_{s=1}^{k} s$ gives us

$$\frac{\sum_{s=1}^{k} s\mathbb{E}[S(x_s, \lambda_s)] - \sum_{s=1}^{k} s\mathbb{E}[S(x_*, \lambda_s)]}{\sum_{s=1}^{k} s} \;\leq\; \frac{kM}{c\sum_{s=1}^{k} s} \;\leq\; \frac{2M}{c(k+1)}. \tag{24}$$

The left-hand side is taken care as follows

$$\min_{s=1,\ldots,k} \mathbb{E}[S(x_s,\lambda_s)] - \mathbb{E}[S(x_*,\bar{\lambda}_k)] \;\leq\; \sum_{s=1}^{k} \frac{s}{\sum_{s=1}^{k} s}\mathbb{E}[S(x_s,\lambda_s)] - \sum_{s=1}^{k} \frac{s}{\sum_{s=1}^{k} s}\mathbb{E}[S(x_*,\lambda_s)], \tag{25}$$

where $\bar{\lambda}_k = \sum_{s=1}^{k} \frac{s}{\sum_{s=1}^{k} s}\lambda_s$. The proof is finally completed by combining (24) and (25). □

Since the sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ generated by Algorithm 1 is bounded, it has a limit point λ_*. Assume that the whole sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ converges to λ_*. Let x_* be the unique minimizer of $S(x,\lambda_*)$. Then, x_* is a Pareto minimizer associated with λ_*. Since $\bar{\lambda}_k$ is also converging to λ_*, $\mathbb{E}[S(x_*,\bar{\lambda}_k)]$ converges to $\mathbb{E}[S(x_*,\lambda_*)]$. Hence, Theorem 5.1 states that $\min_{1\leq s\leq k}\mathbb{E}[S(x_s,\lambda_s)]$ converges to $\mathbb{E}[S(x_*,\lambda_*)]$. The result of Theorem 5.1 indicates that the approximate rate of such convergence is $1/k$. Rigorously speaking, since we do not have λ_* on the left-hand side but rather $\bar{\lambda}_k$, such left-hand side is not even guaranteed to be positive. The difficulty comes from the fact that λ_* is only defined at convergence and the multi-gradient method cannot anticipate which optimal weights are being approached, or in equivalent words which weighted function is being minimized at the end. Such a difficulty is resolved if we assume that λ_k approximates well the role of λ_* at the Pareto front.

Assumption 5.6 *Let x_* be the Pareto minimizer defined above. For any x_k, one has*

$$\nabla_x S(x_*,\lambda_k)^\top (x_k - x_*) \;\geq\; 0.$$

In fact notice that $\nabla_x S(x_*,\lambda_*) = 0$ holds according to the Pareto stationarity condition (9), and thus this assumption would hold with λ_k replaced by λ_*.

A well-known equivalent condition to (20) is

$$(\nabla_x S(x,\lambda) - \nabla_x S(\bar{x},\lambda))^\top (x-\bar{x}) \;\geq\; c\|x-\bar{x}\|^2, \quad \forall (x,\bar{x}) \;\in\; \mathbb{R}^n\times\mathbb{R}^n.$$

Choosing $x = x_k$, $\bar{x} = x_*$, and $\lambda = \lambda_k$ in the above inequality and using Assumption 5.6 leads to

$$\nabla_x S(x_k,\lambda_k)^\top (x_k - x_*) \;\geq\; c\|x_k - x_*\|^2, \tag{26}$$

based on which one can derive a stronger convergence result[2].

Theorem 5.2 *Let Assumptions 5.1–5.6 hold and x_* be the Pareto minimizer corresponding to the limit point λ_* of the sequence $\{\lambda_k\}$. Consider a diminishing step size sequence $\alpha_k = \frac{\gamma}{k}$ where $\gamma \geq \frac{1}{2c}$ is a positive constant. The sequence of iterates generated by Algorithm 1 satisfies*

$$\mathbb{E}[\|x_k - x_*\|^2] \;\leq\; \frac{\max\{2\gamma^2\bar{M}^2(2c\gamma-1)^{-1}, \|x_0 - x_*\|^2\}}{k},$$

and

$$\mathbb{E}[S(x_k,\lambda_*)] - \mathbb{E}[S(x_*,\lambda_*)] \;\leq\; \frac{(L/2)\max\{2\gamma^2\bar{M}^2(2c\gamma-1)^{-1}, \|x_0 - x_*\|^2\}}{k}$$

where $\bar{M} = L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)$.

[2] Let us see how Assumption 5.6 relates to Assumption H5 used in [37]. These authors have made the strong assumption that the noisy values satisfy a.s. $f_i(x,w) - f_i(x,w) \geq C_i\|x - x^\perp\|^2$ for all x, where $x^\perp$ is the point in $\mathcal{P}$ closest to x (and C_i a positive constant). From here they easily deduce from the convexity of the individual functions f_i that $\mathbb{E}_{w_k}[g(x_k,w_k)]^\top(x_k - x_k^\perp) \geq 0$, which then leads to establishing that $\mathbb{E}[\|x_k - x_k^\perp\|^2] = \mathcal{O}(1/k)$.

Notice that $\mathbb{E}_{w_k}[g(x_k,w_k)]^\top(x_k - x_k^\perp) \geq 0$ would also result from (26) (with x_* replaced by $x_k^\perp$) if $g(x_k,w_k)$ was an unbiased estimator of $\nabla_x S(x_k,\lambda_k)$.

Proof. Similarly to the proof of Theorem 5.1, from (21) to (22), but using (26) instead of (23), one has

$$\mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] \;\leq\; (1 - 2\alpha_k c)\|x_k - x_*\|^2 + \alpha_k^2(L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)).$$

Taking total expectation on both sides leads to

$$\mathbb{E}[\|x_{k+1} - x_*\|^2] \;\leq\; (1 - 2\alpha_k c)\mathbb{E}[\|x_k - x_*\|^2] + \alpha_k^2 \bar{M}.$$

Using $\alpha_k = \gamma/k$ with $\gamma > 1/(2c)$ and an induction argument (see [33, Eq. (2.9) and (2.10)]) would lead us to

$$\mathbb{E}[\|x_k - x_*\|^2] \;\leq\; \frac{\max\{2\gamma^2 \bar{M}^2(2c\gamma - 1)^{-1}, \|x_0 - x_*\|^2\}}{k}.$$

Finally, from an expansion using the Lipschitz continuity of $\nabla_x S(\cdot, \lambda_*)$ (see (12)), one can also derive a sublinear rate in terms of the optimality gap of the weighted function value

$$\begin{aligned}\mathbb{E}[S(x_k, \lambda_*)] - \mathbb{E}[S(x_*, \lambda_*)] &\leq \mathbb{E}[\nabla_x S(x_*, \lambda_*)]^\top (x_k - x_*) + \frac{L}{2}\mathbb{E}[\|x_k - x_*\|^2] \\ &\leq \frac{(L/2)\max\{2\gamma^2 \bar{M}^2(2c\gamma - 1)^{-1}, \|x_0 - x_*\|^2\}}{k}.\end{aligned}$$

□

5.2 The convex case

In this section, we relax the strong convexity assumption to convexity and derive a similar sublinear rate of $1/\sqrt{k}$ in terms of weighted function value. Similarly to the case $m = 1$, we assume that the weighted functions attains a minimizer (which then also ensures that $\mathcal{P}$ is non empty).

Assumption 5.7 *All the objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex, $i = 1, \ldots, m$. The convex function $S(\cdot, \lambda)$ attains a minimizer for any $\lambda \in \Delta^m$.*

Theorem 5.3 ***(sublinear convergence rate under convexity)*** *Let Assumptions 5.1–5.4 and 5.7 hold and x_* be any point in $\mathcal{X}$. Consider a diminishing step size sequence $\alpha_k = \frac{\bar{\alpha}}{\sqrt{k}}$ where $\bar{\alpha}$ is any positive constant. The sequence of iterates generated by Algorithm 1 satisfies*

$$\min_{s=1,\ldots,k} \mathbb{E}[S(x_s, \lambda_s)] - \mathbb{E}[S(x_*, \bar{\lambda}_k)] \;\leq\; \frac{\frac{\Theta^2}{2\bar{\alpha}} + \bar{\alpha}(L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S))}{\sqrt{k}},$$

where $\bar{\lambda}_k = \frac{1}{k}\sum_{s=1}^k \lambda_s \in \Delta^m$.

Proof. See Appendix A. □

Similar comments and analysis as in the strongly convex (see the last part of Subsection 5.1 after the proof of Theorem 5.1) could be here derived for the convex case. When comparing to the strongly convex case, we point out that not only the rate is worse in the convex case (as happens also when $m = 1$) but also $\bar{\lambda}_k$ is now converging slower to λ_*.

5.3 Imposing a bound on the biasedness of the multi-gradient

Recall that from

$$\|\mathbb{E}_w\left[g(x,w) - \nabla_x S(x,\lambda^g)\right]\| \;\leq\; \sum_{i=1}^{m} \mathbb{E}_w\left[\|g_i(x,w) - \nabla f_i(x)\|\right], \tag{27}$$

where $g_i(x,w)$ is the stochastic gradient at x for the i-th objective function, and from Assumption 5.2 (b), we derived a more general bound for the biasedness of the stochastic multi-gradient in (14), whose right-hand side involves the stepsize α_k. For simplicity, we will again omit the index k in the subsequent analysis.

We will see that the right-hand side of (27) can be always (approximately) bounded by a dynamic sampling strategy when calculating the stochastic gradients for each objective function. The idea is similar to mini-batch stochastic gradient, in the sense that by increasing the batch size the noise is reduced and thus more accurate gradient estimates are obtained.

Assumption 5.2 (a) states that $g_i(x,w), i = 1,\ldots,m$, are unbiased estimates of the corresponding true gradients. Let us assume that $g_i(x,w)$ is normally distributed with mean $\nabla f_i(x)$ and variance σ_i^2, i.e., $g_i(x,w) \sim \mathcal{N}(\nabla f_i(x), \sigma_i^2 I_n)$, where n is the dimension of x. For each objective function, one can obtain a more accurate stochastic gradient estimate by increasing the batch size. Let b_i be the batch size for the i-th objective function and $\bar{g}_i(x,w) = \frac{1}{b_i}\sum_{r=1}^{b_i} g_i(x,w^r)$ be the corresponding batch stochastic gradient, where $\{w^r\}_{1\leq r\leq b_i}$ are drawn from copies of w. Then, $G_i = g_i(x,w) - \nabla f_i(x)$ and $\bar{G}_i = \bar{g}_i(x,w) - \nabla f_i(x), i = 1,\ldots,m$, are all random variables of mean 0. The relationship between G_i and $\bar{G}_i$ is captured by (see [25])

$$\mathbb{V}_w[\bar{G}_i] \;\leq\; \frac{\mathbb{V}_w[G_i]}{b_i} \;\leq\; \frac{\sigma_i^2}{b_i}.$$

By the definition of variance $\mathbb{V}_w[G_i] = \mathbb{E}_w[\|G_i\|^2] - \|\mathbb{E}_w[G_i]\|^2$, $\|\mathbb{E}_w[\bar{G}_i]\| \leq \mathbb{E}_w[\|\bar{G}_i\|]$, and $\mathbb{E}_w[\bar{G}_i] = 0$, one has $\mathbb{V}_w[\|\bar{G}_i\|] \leq \mathbb{V}_w[\bar{G}_i] \leq \sigma_i^2/b_i$. Then, replacing $g_i(x,w)$ in (27) by $\bar{g}_i(x,w)$, we have

$$\|\mathbb{E}_w\left[\bar{g}(x,w) - \nabla_x S(x,\lambda^g)\right]\| \;\leq\; \sum_{i=1}^{m} \mathbb{E}_w\left[\|\bar{G}_i\|\right] \;\leq\; \sum_{i=1}^{m} \frac{\sigma_i\sqrt{n}}{b_i},$$

where the last inequality results from $\mathbb{E}[\|X\|] \leq \sigma\sqrt{n}$ for the random variable $X \sim \mathcal{N}(0,\sigma^2 I_n)$ [9]. Hence, one could enforce an inequality of the form $\sum_{i=1}^{m} \frac{\sigma_i\sqrt{n}}{b_i} \leq \alpha_k(M_1 + M_F \sum_{i=1}^{m} \|\nabla f_i(x)\|)$ to guarantee that (14) holds (of course replacing the size of the true gradients by some positive constant). Furthermore, to guarantee that the stronger bound (13) holds, one can require $\mathbb{E}_w[\|\bar{G}_i\|] \leq \frac{\sigma_i\sqrt{n}}{b_i} \leq \alpha(C_1 + \hat{C}_i\|\nabla f_i(x)\|)$ for each objective function. Intuitively, when smaller stepsizes are taken, the sample sizes $\{b_i\}_{1\leq i\leq m}$ should be increased or, correspondingly, smaller sample variances $\{\sigma_i\}_{1\leq i\leq m}$ should be used.

6 Pareto-front stochastic multi-gradient method

The practical goal in many MOO problems is to calculate a good approximation of part of (or the entire) Pareto front, and for this purpose the SMG algorithm is insufficient as running it only yields a single Pareto stationary point. We will thus design a Pareto-Front Stochastic Multi-Gradient (PF-SMG) algorithm to obtain the complete Pareto front. The key idea of such an algorithm is to iteratively update a list of nondominated points which will render increasingly better approximations to the true Pareto front. The list is updated by essentially applying the SMG algorithm at some or all of its current points. The PF-SMG algorithm begins with a list of (possibly random) starting points $\mathcal{L}_0$. At each iteration, before applying SMG and for sake of better performance, we first add to the list a certain number, say r, of perturbed points around each of the current ones. Then we apply a certain number of steps, say p, of SMG at each point in the list, adding each resulting final point to the list. The iteration is finished by removing all dominated points from the list. The PF-SMG algorithm is formally described as follows.

Algorithm 2 Pareto-Front Stochastic Multi-Gradient (PF-SMG) Algorithm

```
 1: Generate a list of starting points $\mathcal{L}_0$. Select $r, p, q \in \mathbb{N}$.
 2: for $k = 0, 1, \ldots$ do
 3:     Set $\mathcal{L}_{k+1} = \mathcal{L}_k$.
 4:     for each point $x$ in the list $\mathcal{L}_{k+1}$ do
 5:         for $t = 1, \ldots, r$ do
 6:             Add the new point $x + w^t$ to the list $\mathcal{L}_{k+1}$ where $w^t$ is a realization of $w_k$.
 7:         end for
 8:     end for
 9:     for each point $x$ in the list $\mathcal{L}_{k+1}$ do
10:         for $t = 1, \ldots, p$ do
11:             Apply $q$ iterations of the SMG algorithm starting from $x$.
12:             Add the final output point $x_q$ to the list $\mathcal{L}_{k+1}$.
13:         end for
14:     end for
15:     Remove all the dominated points from $\mathcal{L}_{k+1}$.
16: end for
```

In order to evaluate the performance of the PF-SMG algorithm and have a good benchmark for comparison, we also introduce a Pareto-front version of the deterministic multi-gradient algorithm (acronym PF-MG). The PF-MG algorithm is exactly the same as the PF-SMG one except that one applies q steps of multi-gradient descent instead of stochastic multi-gradient to each point in Line 11. Also, p is always equal to one in PF-MG.

7 Numerical experiments

7.1 Parameter settings and metrics for comparison

In our implementation, both PF-SMG and PF-MG algorithms use the same 30 randomly generated starting points, i.e., $|\mathcal{L}_0| = 30$. In both cases we set $q = 2$. The step size is initialized differently according to the problem but always halved every 200 iterations. Both algorithms are terminated when either the number of iterations exceeds 1000 or the number of points in the iterate list reaches 1500.

To avoid the size of the list growing too fast, we only generate the r perturbed points for pairs of list points corresponding to the m largest holes along the axes f_i, $i = 1, \ldots, m$. More specifically, given the current list of nondominated points, their function values in terms of f_i

are first sorted in an increasing order, for $i = 1, \ldots, m$. Let $d_i^{j,j+1}$ be the distance between points j and $j+1$ in f_i. Then, the pair of points corresponding to the largest hole along the axis f_i is $(j_i, j_i + 1)$, where $j_i = \operatorname{argmax}_j d_i^{j,j+1}$.

Given the fact that applying the SMG algorithm multiple times to the same point results in different output points, whereas this is not the case for multi-gradient descent, we take $p = 2$ for PF-SMG but let $p = 1$ for PF-MG. Then, we choose $r = 5$ for PF-SMG and $r = 10$ for PF-MG, such that the number of new points added to the list is the same at each iteration of the two algorithms.

To analyze the numerical results, we consider two types of widely-used metrics to measure and compare Pareto fronts obtained from different algorithms, Purity [3] and Spread [14], whose mathematical formula are briefly recalled in Appendix B. In what concerns the Spread metric, we use two variants, maximum size of the holes and point spread, respectively denoted by Γ and Δ.

7.2 Supervised machine learning (logistic regression)

7.2.1 Problem description

The idea to construct a multi-objective problem for binary classification problems is inspired by the existence of bias in real data sets, as data instances may be collected for the same classification problem but actually from distinct distributions. Some related issues, like the fairness concern, were addressed in [5]. In fact, if one has a data set collected from different sources or groups, it is necessary to do classification separately such that the accuracy is higher for all groups. However, sometimes we may collect data from distinct groups but cannot determine the existence of bias. Designing a multi-objective formulation might help us identifying if there exists bias and define the best trade off if it does exist.

We have tested our idea on classical binary classification problems with the training data sets selected from LIBSVM [10]. Each data set consists of feature vectors and labels for a number of data instances. The goal of binary classification is to fit the best prediction hyperplane in order to well classify the set of data instances into two groups. More precisely, for a given pair of feature vector a and label y, we consider a separating hyperplane $x^\top a + b$ such that

$$\begin{cases} x^\top a + b \ \geq \ 1 & \text{when } y \ = \ 1, \\ x^\top a + b \ \leq \ -1 & \text{when } y \ = \ -1. \end{cases}$$

In our context, we evaluate the prediction loss using the smooth convex logistic function $l(a, y; x, b) = \log(1+\exp(-y(x^\top a+b)))$, which leads us to a well-studied convex objective, i.e., logistic regression problem with the objective function being $\min_{x,b} \frac{1}{N} \sum_{j=1}^{N} \log(1+\exp(-y_j(x^\top a_j + b)))$, where N is the size of training data. To avoid over-fitting, we need to add a regularization term $\frac{\lambda}{2}\|x\|^2$ to the objective function.

For the purpose of our study, we pick a feature of binary values and separate the given data set into two groups, with J_1 and J_2 as their index sets. An appropriate two-objective problem is formulated as $\min_{x,b} \ (f_1(x, b), f_2(x, b))$, where

$$f_i(x, b) = \frac{1}{|J_i|} \sum_{j \in J_i} \log(1 + e^{(-y_j(x^\top a_j + b))}) + \frac{\lambda_i}{2}\|x\|^2. \tag{28}$$

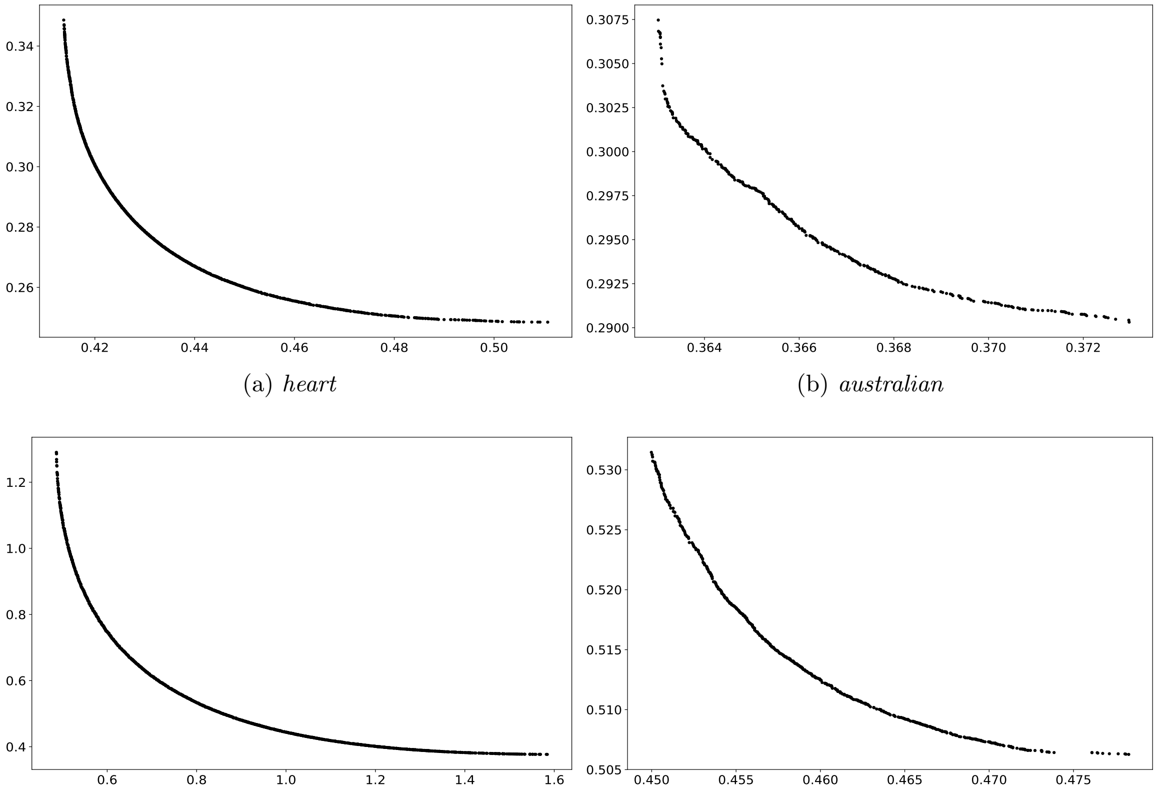
7.2.2 Numerical results

Our numerical results are constructed for four data sets: *heart*, *australian*, *svmguide3*, and *german.numer* [10]. First of all, we ran the single stochastic gradient (SG) algorithm with maximum 1000 iterations to obtain a minimizer for the entire group, i.e., $J_1 \cup J_2$, based on which the classification accuracies for the whole group and for two groups separately are calculated. See Table 1 for a summary of the results. "Split" indicates which feature is selected to split the whole group into two distinct groups. The initial step size is also listed in the table.

Data	N	Step size	Split	Group 1	Group 2	Entire group
heart	270	0.2	2	0.475	0.770	0.570
australian	690	0.3	1	0.774	0.829	0.791
svmguide3	1243	0.2	10	0.794	0.28	0.761
german.numer	1000	0.1	24	0.595	0.530	0.571

Table 1: Classification accuracy of the single SG.

One can easily observe that there exist obvious differences in term of the training accuracy between the two groups of data sets *heart* and *svmguide3*, whereas *australian* and *german.numer* have much smaller gaps. This means that classifying a new instance using the minimizer obtained from the single SG for the whole group might lead to large bias and poor accuracy. We then constructed a two-objective problem (28) with $\lambda_1 = \lambda_2 = 0.1$ for the two groups of each data set. The PF-SMG algorithm has yielded the four Pareto fronts displayed in Figure 3.



(a) *heart*

(b) *australian*

(c) *svmguide3*

(d) *german.numer*

Figure 3: The approximated Pareto fronts for the logistic regression problems: (a) *heart.*; (b) *australian,*; (c) *svmguide3.*; (d) *german.numer..*

The wider Pareto fronts of data sets *heart* and *svmguide3* coherently indicate higher distinction between their two groups. Table 3 presents the number of iterations and the size of Pareto front solutions when the PF-SMG algorithm is terminated. To illustrate the trade-offs, five representative points are selected from the obtained Pareto front, and the corresponding training accuracies are evaluated for the two groups separately. Despite no algorithm comparison here, we also calculated the maximum size of the holes Γ and the point spread Δ for these Pareto fronts.

Data	#Iter	$\lvert\mathcal{L}_k\rvert$	N	P_1	P_2	P_3	P_4	P_5	Γ	Δ
heart	764	1501	183	0.645	0.628	0.607	0.568	0.541	0.0015	0.8974
			87	0.609	0.689	0.736	0.781	0.805		
australian	1000	568	468	0.760	0.756	0.752	0.746	0.737	0.0025	0.9106
			222	0.802	0.797	0.792	0.819	0.829		
svmguide3	116	1515	1182	0.695	0.507	0.148	0.201	0.206	0.0191	0.9905
			61	0.098	0.229	0.656	0.868	0.869		
german.numer	1000	588	630	0.517	0.516	0.508	0.503	0.498	0.0025	0.8389
			370	0.435	0.446	0.446	0.448	0.451		

Table 2: Classification accuracy corresponding to several Pareto minimizers.

It is observed for the groups of data sets *heart* and *svmguide3* that the differences of training accuracy vary more than 10 percent among Pareto minimizers. Two important implications from the results are: (1) Given several groups of data instances for the same problem, one can evaluate their biases by observing the range of an approximated Pareto front; (2) Given a well-approximated Pareto front, any new data instance (of unknown group) can be classified more accurately by selecting appropriate nondominated solutions.

7.3 Synthetic MOO test problems

7.3.1 Test problems

There exist more than a hundred of deterministic MOO problems reported in the literature (involving simple bound constraints), and they were collected in [12]. Our testing MOO problems are taken from this collection (see [12] for the problem sources) and include four cases: convex, concave, mixed (neither convex nor concave), and disconnected Pareto fronts. Table 3 provides relevant information including number of objectives, variable dimensions, simple bounds, and geometry types of Pareto fronts for the 13 selected MOO problems.

Problem	n	m	Simple bounds	Geometry
ZDT1	30	2	$[0,1]$	convex
ZDT2	30	2	$[0,1]$	concave
ZDT3	30	2	$[0,1]$	disconnected
JOS2	10	2	$[0,1]$	mixed
SP1	2	2	No	convex
IM1	2	2	$[1,4]$	concave
FF1	2	2	No	concave
Far1	2	2	$[-1,1]$	mixed
SK1	1	2	No	disconnected
MOP1	1	2	No	convex
MOP2	15	2	[-4, 4]	concave
MOP3	2	2	$[-\pi,\pi]$	disconnected
DEB41	2	2	$[0,1]$	convex

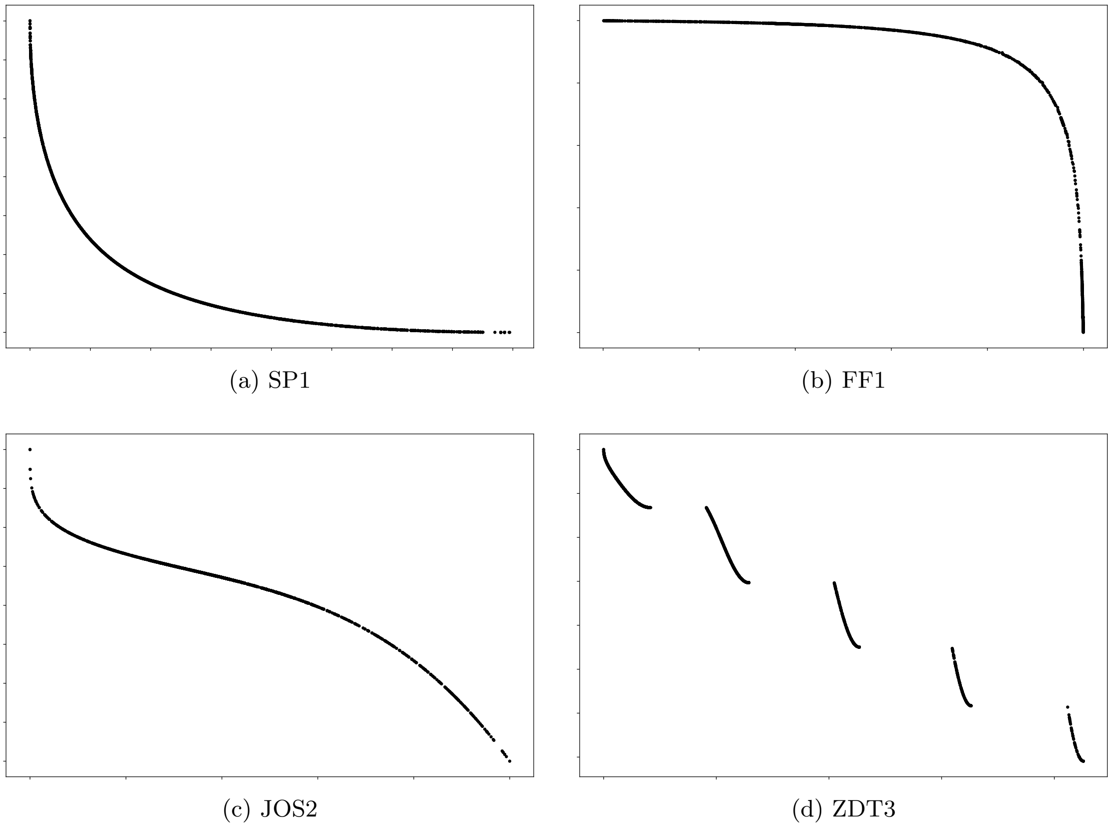
Table 3: 13 MOO testing problems.

The way to construct a corresponding stochastic MOO problem from its deterministic MOO problem was the same as in [37]. For each of these MOO test problems, we added random noise to its variables to obtain a stochastic MOO problem, i.e.,

$$\begin{array}{ll} \min & F(x) = (\mathbb{E}[f_1(x+w)], \ldots, \mathbb{E}[f_m(x+w)])^\top \\ \text{s.t.} & x \in \mathcal{X}, \end{array}$$

where w is uniformly distributed with mean zero and interval length being 1/10 of the length of the simple bound interval (the latter one was artificially chosen when not given in the problem description). Note that the stochastic gradients will not be unbiased estimates of the true gradients of each objective function, but rather gradients of randomly perturbed points in the neighborhood of the current point.

Figure 4 illustrates four different geometry shapes of Pareto fronts obtained by removing all dominated points from the union of the resulting Pareto fronts obtained from the application of the PF-SMG and PF-MG algorithms. In the next subsection, the quality of approximated Pareto fronts obtained from the two algorithms is measured and compared in terms of the Purity and Spread metrics.



(a) SP1

(b) FF1

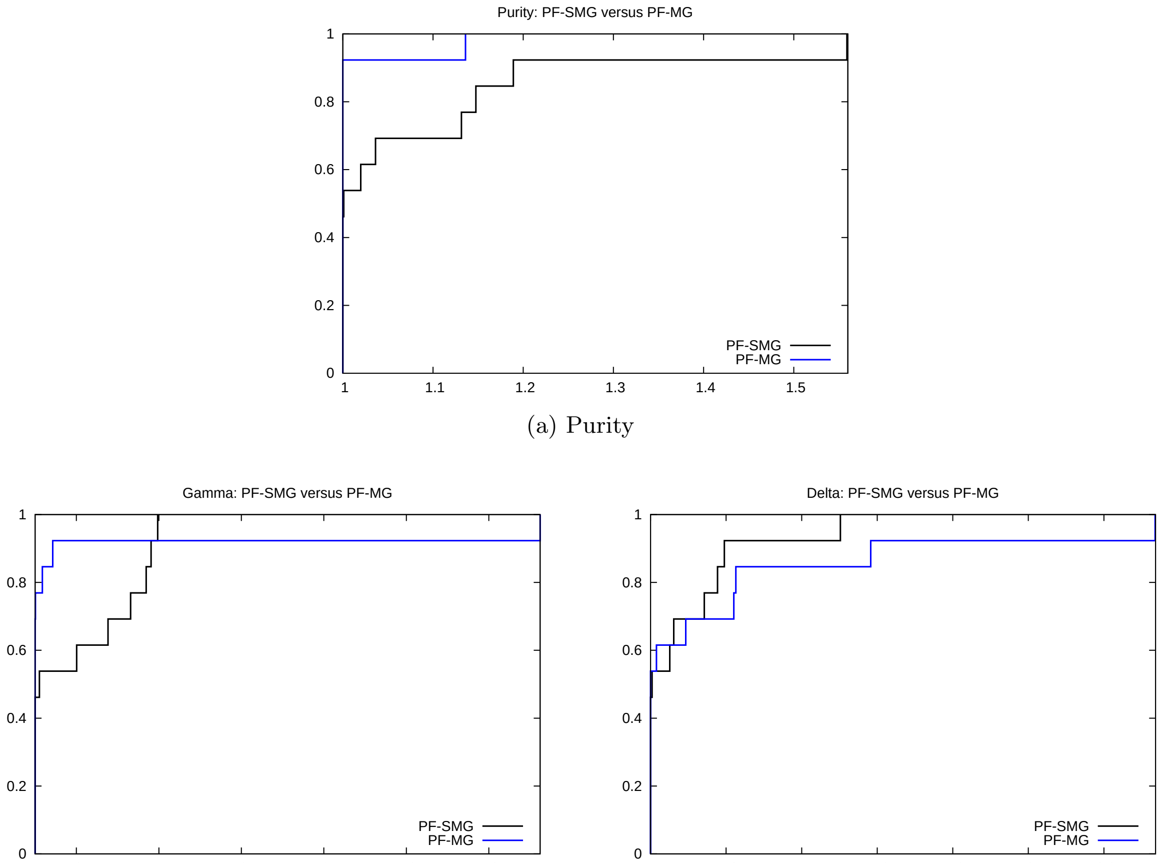
(c) JOS2

(d) ZDT3

Figure 4: Different geometry shapes of Pareto fronts: (a) Convex; (b) Concave; (c) Mixed (neither convex nor concave); (d) Disconnected.

7.3.2 Numerical results and analysis

For all problems, the initial step size was set to 0.3 for both PF-SMG and PF-MG algorithms. To be fair, we ran 10 times the PF-SMG algorithm for each problem and selected the one with the average value in Γ, i.e., the maximum size of the holes. (Although there is some randomization in PF-MG, its output does not differ significantly from one run to another.) The quality of the obtained Pareto fronts, the number of iterations, and the size of the Pareto front approximations when the algorithms are terminated are reported in Table 4. We also plot performance profiles, see Figure 5, in terms of Purity and the two formula of Spread metrics.



(a) Purity

(b) Spread: Γ

(c) Spread: Δ

Figure 5: Performance profiles in terms of Purity, Γ, and Δ repectively.

Overall, the PF-MG algorithm produces Pareto fronts of higher Purity than the PF-SMG, which is reasonable since using the accurate gradient information results in points closer to the true Pareto front. However, the Purity of Pareto fronts resulting from the PF-SMG is quite close to the one from the PF-MG in most of the testing problems. Also, when we examine the quality of the fronts in terms of the Spread metrics (see Γ and Δ in Table 4), their performances are comparable, which indicates that the proposed PF-SMG algorithm is able to produce well-spread Pareto fronts. For some problems like IM1 and FF1, it is observed that PF-SMG generates nondominated points faster than the PF-MG. This might be due to the fact PF-SMG has two sources of stochasticity, both in generating the points and in applying stochastic multi-gradient, whereas PF-MG is only stochastic in the generation of points.

On the other hand, perhaps due to the worse accuracy of stochastic multi-gradients, PF-SMG takes more iterations than PF-MG to achieve the same tolerance level. Nevertheless, suppose that the computational cost for computing the true gradients for each objective function is significantly higher than the one for obtaining the stochastic gradients. It is easy to consider scenarios when the computation cost of PF-MG would be far more expensive than for PF-SMG.

Problem	Algorithm	Purity	Γ	Δ	# Iter	$\lvert\mathcal{L}_k\rvert$
ZDT1	PF-MG	1.000	0.0332	1.4404	26	1575
	PF-SMG	1.000	0.0666	1.6958	26	1789
ZDT2	PF-MG	1.000	0.9336	1.0407	48	1524
	PF-SMG	1.000	0.0705	1.5637	32	1680
ZDT3	PF-MG	0.999	0.1716	1.5941	84	1524
	PF-SMG	0.999	0.6539	1.3005	70	1544
JOS2	PF-MG	1.000	0.1853	1.3520	24	1530
	PF-SMG	1.000	0.7358	1.5445	18	2271
SP1	PF-MG	0.996	0.0763	1.5419	24	1826
	PF-SMG	0.880	0.2817	0.9742	102	1503
IM1	PF-MG	0.992	0.0936	0.8879	18	1581
	PF-SMG	0.973	0.2591	1.0613	16	2161
FF1	PF-MG	0.982	0.0788	1.5637	46	1533
	PF-SMG	0.630	0.0671	1.5701	20	1834
Far1	PF-MG	0.843	0.3800	1.5072	26	1741
	PF-SMG	0.958	0.4192	1.5996	44	1602
SK1	PF-MG	1.000	24.6399	1.0053	68	1531
	PF-SMG	0.999	24.6196	0.9195	48	1614
MOP1	PF-MG	1.000	0.0329	0.9003	78	1505
	PF-SMG	1.000	0.1091	0.9462	14	2036
MOP2	PF-MG	1.000	0.0614	1.8819	140	1527
	PF-SMG	0.841	0.0609	0.8057	124	1504
MOP3	PF-MG	0.990	19.8772	1.7938	26	1530
	PF-SMG	0.863	19.8667	1.7664	50	1571
DEB41	PF-MG	0.953	26.8489	1.8430	14	1813
	PF-SMG	0.920	18.8147	1.5101	18	1997

Table 4: Comparison between resulting Pareto fronts from the PF-MG and PF-SMG algorithms.

Two informative final notes. The Pareto fronts of problem SK1 and MOP3 are disconnected, and hence, their values of Γ are significantly larger than others. There exists a conflict between depth (Purity) and breadth (Spread) of the Pareto front. One can always tune some parameters, e.g., the number of starting points and the number of points generated per point at each iteration, to balance the Purity and Spread of the resulting Pareto fronts.

8 Conclusions

The stochastic multi-gradient (SMG) method is an extension of the stochastic gradient method from single to multi-objective optimization (MOO). However, even based on the assumption of unbiasedness of the stochastic gradients of the individual functions, it has been observed in this paper that there exists a bias between the stochastic multi-gradient and the corresponding true multi-gradient, essentially due to the composition with the solution of a quadratic program (see (10)). Imposing a condition on the amount of tolerated biasedness, we established sublinear convergence rates, $\mathcal{O}(1/k)$ for strongly convex and $\mathcal{O}(1/\sqrt{k})$ for convex objective functions, similar to what is known for single-objective optimization, except that the optimality gap was measured in terms of a weighted sum of the individual functions. We realized that the main difficulty in establishing these rates for the multi-gradient method came from the unknown limiting behavior of the weights generated by the algorithm. Nonetheless, our theoretical results contribute to a deeper understanding of the convergence rate theory of the classical stochastic gradient method in the MOO setting.

To generate the entire Pareto front in a single run, the SMG algorithm was framed into a Pareto-front one, iteratively updating a list of nondominated points. The resulting PF-SMG algorithm was shown to be a robust technique for smooth stochastic MOO since it has produced well-spread and sufficiently accurate Pareto fronts, while being relatively efficient in terms of the overall computational cost. Our numerical experiments on binary logistic regression problems showed that solving a well-formulated MOO problem can be a novel tool for identifying biases among potentially different sources of data and improving the prediction accuracy.

As it is well known, noise reduction [15, 30, 35, 40] was studied intensively during the last decade to improve the performance of the stochastic gradient method. Hence, a relevant topic for our future research is the study of noise reduction in the setting of the stochastic multi-gradient method for MOO. More applications and variants of the algorithm can be further explored. For example, we have not yet tried to solve stochastic MOO problems when the feasible region is different from box constraints. We could also consider the incorporation of a proximal term and in doing so we could handle nonsmooth regularizers. Other models arising in supervised machine learning, such as the deep learning, could be also framed into an MOO context. Given that the neural networks used in deep learning give rise to nonconvex objective functions, we would also be interested in developing the convergence rate theory for the SMG algorithm in the nonconvex case.

A Proof of Theorem 5.3

Proof. By applying inequalities (16), (17), and (19) to (22), one obtains

$$\begin{aligned} \mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] \;&\leq \|x_k - x_*\|^2 + \alpha_k^2(L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)) \\ &\quad 2\alpha_k \mathbb{E}_{w_k}[\nabla_x S(x_k, \lambda_k)]^\top (x_k - x_*). \end{aligned} \tag{29}$$

From convexity one has

$$S(x_k, \lambda_k) - S(x_*, \lambda_k) \;\leq\; \nabla_x S(x_k, \lambda_k)^\top (x_k - x_*). \tag{30}$$

Then, plugging (30) into inequality (29) and rearranging yield

$$\begin{aligned} 2\alpha_k(\mathbb{E}_{w_k}[S(x_k, \lambda_k)] - \mathbb{E}_{w_k}[S(x_*, \lambda_k)]) \;&\leq \|x_k - x_*\|^2 - \mathbb{E}_{w_k}[\|x_{k+1} - x_*\|^2] \\ &\quad + \alpha_k^2(L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)). \end{aligned}$$

For simplicity denote $\hat{M} = L_g^2 + 2\Theta(L_{\nabla S} + \beta L_{\nabla S} M_S)$. Dividing both sides by α_k and taking total expectations on both sides allow us to write

$$2(\mathbb{E}[S(x_k, \lambda_k)] - \mathbb{E}[S(x_*, \lambda_k)]) \;\leq\; \frac{\mathbb{E}[\|x_k - x_*\|^2] - \mathbb{E}[\|x_{k+1} - x_*\|^2]}{\alpha_k} + \alpha_k \hat{M}.$$

Replacing k by s in the above inequality and summing over $s = 1, \dots, k$ lead to

$$\begin{aligned} 2\sum_{s=1}^{k}(\mathbb{E}[S(x_s, \lambda_s)] - \mathbb{E}[S(x_*, \lambda_s)]) &\le \frac{1}{\alpha_1}\mathbb{E}[\|x_1 - x_*\|^2] + \sum_{s=1}^{k}\alpha_s\hat{M} \\ &\quad + \sum_{s=2}^{k}(\frac{1}{\alpha_s} - \frac{1}{\alpha_{s-1}})\mathbb{E}[\|x_s - x_*\|^2] \\ &\le \frac{\Theta^2}{\alpha_1} + \sum_{s=2}^{k}(\frac{1}{\alpha_s} - \frac{1}{\alpha_{s-1}})\Theta^2 + \sum_{s=1}^{k}\alpha_s\hat{M} \\ &\le \frac{\Theta^2}{\alpha_k} + \sum_{s=1}^{k}\alpha_s\hat{M}. \end{aligned}$$

Then, using $\alpha_s = \frac{\bar{\alpha}}{\sqrt{s}}$ and dividing both sides by $2k$ in the last inequality give us

$$\frac{1}{k}\sum_{s=1}^{k}(\mathbb{E}[S(x_s, \lambda_s)] - \mathbb{E}[S(x_*, \lambda_s)]) \;\le\; \frac{\Theta^2}{2\bar{\alpha}\sqrt{k}} + \frac{\bar{\alpha}\hat{M}}{\sqrt{k}}, \tag{31}$$

from the fact $\sum_{s=1}^{k}\frac{\bar{\alpha}}{\sqrt{s}} \le 2\bar{\alpha}\sqrt{k}$. In the left-hand side, one can use the following inequality

$$\min_{s=1,\dots,k} \mathbb{E}[S(x_k, \lambda_k)] - \mathbb{E}[S(x_*, \bar{\lambda}_k)] \;\le\; \frac{1}{k}\sum_{s=1}^{k}(\mathbb{E}[S(x_s, \lambda_s)] - \mathbb{E}[S(x_*, \lambda_s)]), \tag{32}$$

where $\bar{\lambda}_k = \frac{1}{k}\sum_{s=1}^{k}\lambda_s$. The final result follows from combining (31) and (32). □

B Metrics for comparison

Let $\mathcal{A}$ denote the set of algorithms/solvers and $\mathcal{T}$ denote the set of test problems. The Purity metric measures the accuracy of an approximated Pareto front. Let us denote $H(\mathcal{P}_{a,t})$ as an approximated Pareto front of problem t computed by algorithm a. We approximate the true Pareto front $H(\mathcal{P}_t)$ for problem t by all the nondominated solutions in $\cup_{a\in\mathcal{A}}H(\mathcal{P}_{a,t})$. Then, the Purity of a Pareto front computed by algorithm a for problem t is the ratio $r_{a,t} = |H(\mathcal{P}_{a,t}) \cap H(\mathcal{P}_t)|/|H(\mathcal{P}_t)| \in [0, 1]$, which calculates the percentage of nondominated solutions that are common in the approximated Pareto front and the true Pareto front. A higher ratio value corresponds to a more accurate Pareto front. In our context, it is highly possible that the Pareto front obtained from PF-MG algorithm dominates that from the PF-SMG algorithm since the former one uses true gradients.

The Spread metric is designed to measure the extent of the point spread in a computed Pareto front, which requires the computation of extreme points in the objective function space $\mathbb{R}^m$. Among m objective functions, we select a pair of nondominated points with the highest pairwise distance measured using f_i as the pair of extreme points. The first formula calculates the maximum size of the holes for a Pareto front. Assume algorithm a generates an approximated Pareto front with M points, indexed by $1, \dots, M$, to which the pair of extreme points indexed by 0 and $M+1$ are added. Denote the maximum size of the holes by Γ. We have

$$\Gamma \; = \; \Gamma_{a,t} \; = \; \max_{i \in \{0,\ldots,m\}} \left(\max_{j \in \{1,\ldots,N\}} \{\delta_{i,j}\} \right), \tag{33}$$

where $\delta_{i,j} = f_{i,j+1} - f_{i,j}$, and we assume each of the objective function values f_i is sorted in an increasing order.

The second formula was proposed by [14] for the case $m = 2$ (and further extended to the case $m \geq 2$ in [12]) and indicates how well the points are distributed in a Pareto front. Denote the point spread by Δ. It is computed by the following formula:

$$\Delta \; = \; \Delta_{a,t} \; = \; \max_{i \in \{1,\ldots,m\}} \left(\frac{\delta_{i,0} + \delta_{i,M} + \sum_{j=1}^{M-1} |\delta_{i,j} - \bar{\delta}_i|}{\delta_{i,0} + \delta_{i,M} + (M-1)\bar{\delta}_i} \right), \tag{34}$$

where $\bar{\delta}_i, i = 1, \ldots, m$ is the average of $\delta_{i,j}$ over $j = 1, \ldots, M-1$. Note that the lower Γ and Δ are, the more well distributed the Pareto front is.

References

[1] F. B. Abdelaziz. *L'efficacité en Programmation Multi-Objectifs Stochastique.* PhD thesis, Université de Laval, Québec, 1992.

[2] F. B. Abdelaziz. Solution approaches for the multiobjective stochastic programming. *European J. Oper. Res.*, 216:1–16, 2012.

[3] S. Bandyopadhya, S. K. Pal, and B. Aruna. Multiobjective GAs, quantitative indices, and pattern classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34:2088–2099, 2004.

[4] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12:269–283, 2008.

[5] Y. Bechavod, K. Ligett, A. Roth, B. Waggoner, and Z. S. Wu. Equal opportunity in online classification with partial feedback. *arXiv preprint arXiv:1902.02242*, 2019.

[6] H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM J. Optim.*, 15:953–970, 2005.

[7] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60:223–311, 2018.

[8] R. Caballero, E. Cerdá, M. Munoz, and L. Rey. Stochastic approach versus multiobjective approach for obtaining efficient solutions in stochastic multiobjective programming problems. *European J. Oper. Res.*, 158:633–648, 2004.

[9] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Found. Comput. Math.*, 12:805–849, 2012.

[10] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:27, 2011.

[11] K. L. Chung. On a stochastic approximation method. *Ann. Math. Statist.*, 25:463–483, 1954.

[12] A. L. Custódio, J. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. Optim.*, 21:1109–1140, 2011.

[13] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.*, 8:631–657, 1998.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[15] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

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

[17] J. A. Désidéri. Multiple-gradient descent algorithm for Pareto-front identification. In *Modeling, Simulation and Optimization for Science and Technology*, pages 41–58. Springer, Dordrecht, 2014.

[18] L. G. Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Comput. Optim. Appl.*, 28:5–29, 2004.

[19] L. G. Drummond, F. M. P. Raupp, and B. F. Svaiter. A quadratically convergent Newton method for vector optimization. *Optimization*, 63:661–677, 2014.

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

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

[22] J. Fliege, L. G. Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM J. Optim.*, 20:602–626, 2009.

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

[24] J. Fliege, A. I. F. Vaz, and L. N. Vicente. Complexity of gradient descent for multiobjective optimization. *to appear in Optim. Methods Softw.*, 2018.

[25] J. E. Freund. *Mathematical Statistics.* Prentice-Hall, Englewood Cliffs, N.J., 1962.

[26] E. H. Fukuda and L. M. G. Drummond. A survey on multiobjective descent methods. *Pesquisa Operacional*, 34:585–620, 2014.

[27] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Nav. Res. Logist. Q.*, 2:39–45, 1955.

[28] A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.*, 22:618–630, 1968.

[29] Y. V. Haimes. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1:296–297, 1971.

[30] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[31] A. J. Kleywegt, A. Shapiro, and T. Homem de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.*, 12:479–502, 2002.

[32] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, New York, 2012.

[33] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19:1574–1609, 2009.

[34] L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J. Optim.*, 28:2690–2720, 2018.

[35] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30:838–855, 1992.

[36] S. Qu, M. Goh, and B. Liang. Trust region methods for solving multiobjective optimisation. *Optim. Methods Softw.*, 28:796–811, 2013.

[37] M. Quentin, P. Fabrice, and J. A. Désidéri. A stochastic multiple gradient descent algorithm. *European J. Oper. Res.*, 271:808 – 817, 2018.

[38] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951.

[39] J. Sacks. Asymptotic distribution of stochastic approximation procedures. *Ann. Math. Statist.*, 29:373–405, 1958.

[40] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Math. Program.*, 127:3–30, 2011.

[41] A. Shapiro. Monte Carlo sampling methods. *Handbooks in Operations Research and Management Science*, 10:353–425, 2003.

[42] K. D. Villacorta, P. R. Oliveira, and A. Soubeyran. A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. *J. Optim. Theory Appl.*, 160:865–889, 2014.