Stochastic block coordinate and function alternation for multi-objective optimization and learning

T. H. Tran* L. N. Vicente†

May 12, 2026

Multi-objective optimization is central to many engineering and machine learning applications, where multiple objectives must be optimized in balance. While multi-gradient based optimization methods combine these objectives in each step, such methods require computing gradients with respect to all variables at every iteration, resulting in high computational costs in large-scale settings. In this work, we propose a framework that simultaneously alternates the optimization of each objective and the (stochastic) gradient update with respect to each variable block. Our framework reduces per-iteration computational cost while enabling exploration of the Pareto front by allocating a prescribed number of gradient steps to each objective. We establish rigorous convergence guarantees across several stochastic smooth settings, including convex, non-convex, and Polyak-Łojasiewicz conditions, recovering classical convergence rates of single-objective methods. Numerical experiments demonstrate that our framework outperforms non-alternating methods on multi-target regression and produces a competitive Pareto front approximation, highlighting its computational efficiency and practical effectiveness.

1 Introduction

Multi-objective optimization (MOO) arises in numerous real-world applications from engineering to finance where several conflicting criteria must be evaluated simultaneously [17, 25, 36, 37]. Because these objectives potentially compete, the goal is to find a set of Pareto optimal solutions where no single objective can be improved without degrading another. Identifying this set of efficient points enables decision-makers to navigate complex trade-offs effectively. Furthermore, in many modern large-scale applications, these competing objectives are inherently stochastic, and the underlying decision variables exhibit a natural block structure [39, 46, 49]. In this paper, we consider the following problem:

$$\min_{x\in\mathbb{R}^n} F(x) = (f_1(x), \ldots, f_q(x)) = (\mathbb{E}[g_1(x,\xi)], \ldots, \mathbb{E}[g_q(x,\xi)]),$$

where the decision variable $x \in \mathbb{R}^n$ is partitioned into s disjoint blocks $x = (x_1, \ldots, x_s)$, and each block x_i has dimension n_i with $\sum_{i=1}^s n_i = n$. We consider the MOO setting where each individual objective function f_k has a Lipschitz continuous gradient. Moreover, each objective

*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (hht320@lehigh.edu).

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

function is stochastic and computing precise gradients is typically either impossible or requires prohibitive computational resources. To overcome this limitation, our framework assumes access to an unbiased stochastic estimator $\nabla g_k(x, \xi)$ for the true gradient of each f_k, where ξ represents a random variable.

1.1 Literature review and motivation

Existing MOO literature generally divides methods into two main categories: *a priori* and *a posteriori* methods, depending on when the decision-maker's preferences are incorporated into the optimization process. In MOO, the *a priori* approach integrates decision-making preferences beforehand to transform the multi-objective problem into a simpler single-objective one, which can then be solved using single-objective optimization algorithms. Common strategies in this category include the weighted-sum method [21], which assigns non-negative weights to combine the objectives into a single convex linear combination; the ϵ-constrained method [26], which minimizes one primary objective while imposing upper bounds on the others; and the utility function method [6, 37], which employs a scalar function to mathematically quantify the overall preference between different objective vectors. In contrast, *a posteriori* methods are designed to generate a comprehensive or representative set of Pareto optimal solutions from which the decision-maker later selects the best option. These methods typically update a list of candidate solutions using either metaheuristics (such as evolutionary algorithms [2, 10]) or rigorous descent mechanisms that compute common descent directions for all objectives simultaneously [12, 15, 20]. The alternating optimization algorithms proposed in this paper belong, in a certain sense, to both categories.

In smooth, deterministic MOO, the multi-gradient algorithm applies the steepest descent step by computing the minimal-norm convex combination of gradients from individual functions. Therefore, a natural approach for stochastic MOO is the stochastic multi-gradient algorithm [34, 41], where the gradients in the minimal-norm convex combination are replaced by its stochastic estimators. It is shown that the stochastic multi-gradient is a biased estimator of the true multi-gradient of the weighted function, and one can impose a bound on the amount of bias estimation by dynamic sampling [34]. With that assumption, [34] establishes sublinear convergence rates that are similar to the ones known for single-objective optimization: $\mathcal{O}(1/T)$ in the strongly convex case and $\mathcal{O}(1/T^{1/2})$ in the convex one, where T is the number of iterations. While [34] only considers convex and strongly convex settings, follow-up work [8, 18, 19, 54] investigates multi-objective stochastic gradient methods in non-convex case, using different strategies to address the bias in the stochastic multi-gradient. Without variance reduction, [8] shows the convergence rate of $\mathcal{O}(1/T^{1/2})$ for their proposed stochastic algorithm with double sampling in the non-convex case.

In single-objective optimization, block coordinate descent refers to a family of algorithms that update one block of the decision variable at each iteration [3, 32, 46, 49]. This approach is attractive when the variable naturally decomposes into blocks of coordinates and when full-gradient steps are expensive. The update of block variables can be done in either a Gauss-Seidel sequential style or a Gauss-Jacobi parallel style [3, 7, 32]. The order of such block variables can be chosen using a cyclic fashion, a random permutation of all blocks, or a randomized selection, where the new block is chosen uniformly at random. The convergence of deterministic BCD has been analyzed extensively in the literature [3, 24, 35, 39, 42, 48], where [3, 39] consider general smooth convex settings for cyclic BCD and its randomized version, and achieve a convergence

rate of $\mathcal{O}(1/T^{1/2})$. The work [50] pioneered the generalization of BCD for stochastic, single-objective optimization with theoretical results for both convex and non-convex regimes. The update order of the variable blocks can be fixed or shuffled, and the algorithm achieves sub-linear convergence rates for convex and strongly convex cases (namely $\mathcal{O}(1/T^{1/2})$ and $\mathcal{O}(1/T)$, respectively), which match the rates of stochastic gradient descent. However, [50] only shows asymptotic convergence in the non-convex setting. Later work [7, 11, 16, 28, 51, 52] expands this research direction, where [7] provides non-asymptotic convergence guarantees for cyclic BCD methods in non-convex settings, however, utilizing a variance reduction technique.

1.2 Contributions of this paper

In this paper, we develop a stochastic block coordinate and function alternation algorithm (named Block-SMOO) for stochastic MOO. Block-SMOO has a two-loop structure where the outer loop iterates over the blocks of decision variables, while the inner loop applies a specific number of stochastic gradient descent steps for each objective function. Our paper analyzes the theoretical convergence of Block-SMOO in various standard smooth settings, which recovers the classical convergence rates of standard single-objective stochastic gradient algorithms. Notably, as each update of our method only involves one objective function and one block of variables, it incurs a lower computational cost per iteration compared to standard full-gradient approaches like the weighted-sum methods [21, 34].

Block-SMOO differs from existing stochastic MOO literature, where the most closely related work is the stochastic alternating bi-objective algorithm [33]. While [33] only alternates function minimization, our method considers the simultaneous alternation of both functions and variables. Furthermore, [33] focuses on the specific case of two objective functions in convex and strongly convex settings, whereas our theory accommodates a general number of objective functions and provides theoretical guarantees across general convex, non-convex, and Polyak-Łojasiewicz (PL) conditions. Finally, our proof technique fundamentally differs from [33] as we establish convergence by utilizing a recursive descent bound for the objective function value, while the approach in [33] relies on an application of the Intermediate Value Theorem to aggregate the separate optimization steps.

In addition, our paper addresses a critical gap in the block coordinate descent literature for stochastic MOO, as no prior work has analyzed the simultaneous alternation of functions and variable blocks. The closely related BCD work [9] employs scalarization to combine objectives into a single weighted function prior to optimization, rather than treating them separately, but it lacks a theoretical analysis for its proposed method. Our work resolves this by maintaining separated objectives and providing solid theory.

Meanwhile, our numerical experiments demonstrate the practical advantages of our method. When applied to compute an equally weighted Pareto optimal solution, Block-SMOO outperforms standard baseline approaches on multi-target regression tasks. We also validate the Block-SMOO's ability to approximate the full Pareto front, showing that it produces a trade-off surface structurally similar to the traditional weighted-sum approach but with a higher accuracy.

The remainder of this paper is organized as follows. In Section 2, we provide a detailed description of the proposed Block-SMOO algorithm. We then present, in Section 3, the theoretical convergence rate analysis of our method for the smooth, non-convex case. In addition, we derive convergence results for the smooth convex case and the case where the functions satisfy the Polyak-Łojasiewicz (PL) condition. In Section 4, we report the numerical performance of

our algorithm on multi-target regression tasks, demonstrating its practical performance. The paper is concluded with some final remarks in Section 5.

2 Stochastic block coordinate and function alternation algorithm

Described in Algorithm 1, our Block-SMOO operates by applying at each iteration a certain number of stochastic (partial) gradient steps to each individual function, where the order of those steps (meaning the selection of the individual functions) follows a certain user-specified sequence.

Firstly, the algorithm is initialized with a budget of total gradient steps to be applied at each iteration, denoted by p. To specify which of the q individual functions is selected for a gradient step in the order $0, \ldots, p-1$, we introduce an index mapping $\pi : \{0, \ldots, p-1\} \to \{1, \ldots, q\}$. For any gradient step $j \in \{0, \ldots, p-1\}$, the value $\pi(j)$ identifies which individual function is selected for such a step. Next, we introduce a frequency vector $m \in \mathbb{N}^q$, where each component m_k specifies the number of gradient steps that is allocated to the corresponding objective function f_k within a single cycle from 0 to $p-1$. Consequently, the predetermined budget p is also the sum of these individual frequencies, $p = \sum_{k=1}^{q} m_k$. One can then see that the index mapping π is such that the cardinality of its preimage for the objective f_k is m_k, meaning $|\pi^{-1}(k)| = m_k$. This cycle of gradient steps naturally introduces the following weighted-sum function, for which the weights are identified through the vector m:

$$F_m(\cdot) := \sum_{k=1}^{q} \frac{m_k}{p} f_k(\cdot) = \frac{1}{p} \sum_{j=0}^{p-1} f_{\pi(j)}(\cdot),$$

where the second equality holds for every index mapping π due to its definition. The Block-SMOO algorithm aims to minimize this weighted-sum function. To illustrate, consider a bi-objective setting ($q = 2$) with a frequency vector $m = (5, 15)$, corresponding to a budget of $p = 20$. This configuration allocates 25% of the steps to the first objective and 75% to the second. As an example, one valid index mapping π is the contiguous sequence

$$(\underbrace{1, \ldots, 1}_{5 \text{ times}}, \underbrace{2, \ldots, 2}_{15 \text{ times}}),$$

and any arbitrary permutation of this sequence satisfies our algorithmic construction. As a result, the Block-SMOO will minimize the weighted-sum function $F_m = 0.25 f_1 + 0.75 f_2$.

By varying the convex linear weights in F_m, one is guaranteed to find all Pareto solutions when the functions are convex [17, 37]. Therefore, by changing the frequencies m_k for each function f_k in the beginning of Block-SMOO, one can capture the trade-off between all objective functions.

After initialization, for each outer iteration t of Algorithm 1, a permutation σ of the block index set $\{1, \ldots, s\}$ is chosen to randomize the order of the variable blocks. Then, for each selected block $\sigma(i+1)$, it constructs an index mapping π as described above. The algorithm then proceeds to update the chosen block $\sigma(i+1)$ using a stochastic gradient step of the individual function determined by $\pi(j)$. This process is repeated for every mapped sequence of objective ($j = 0, \ldots, p-1$) and every block ($i = 0, \ldots, s-1$), after which the algorithm advances to the

Algorithm 1 Block-SMOO Algorithm

Input: Initial point $x^{0,0,0}$ and a step size sequence $\{\alpha_t\}$.
The budget p and the frequency vector $m \in \mathbb{N}^q$, $p = \sum_{k=1}^{q} m_k$.
for $t = 0, \ldots, T-1$ **do**
  Choose a permutation σ of the block index set $\{1, \ldots, s\}$.
  for $i = 0, \ldots, s-1$ **do**
    Choose an index mapping $\pi : \{0, \ldots, p-1\} \to \{1, \ldots, q\}$ satisfying that $|\pi^{-1}(k)| = m_k$.
    for $j = 0, \ldots, p-1$ **do**
      Generate a stochastic gradient $\nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})$ to estimate $\nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})$.
      Update $x^{t,i,j+1}_{\sigma(i+1)} = x^{t,i,j}_{\sigma(i+1)} - \alpha_t \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})$.
      Set $x^{t,i,j+1}_{k} = x^{t,i,j}_{k}$ for all $k \neq \sigma(i+1)$.
    Set $x^{t,i+1,0} = x^{t,i,p}$.
  Set $x^{t+1,0,0} = x^{t,s,p}$.

next outer iteration. We note that our algorithm and convergence analyses allow for different permutations of σ and π across iterations, however, we will drop the indices t, i throughout our paper for simplicity.

It is worth noting that Block-SMOO generalizes several well-known optimization algorithms. First, in the single-objective case, when $q = 1$, our algorithm naturally reduces to the stochastic Block Coordinate Descent (BCD) method [50]. Second, when $s = 1$, i.e., the decision variable is treated as a single block, the algorithm simplifies to an alternating function method for MOO (matching what was proposed in [33] for two objectives). Also, when $s = 1$, one can see our approach as a way to replicate the final effect of the weighted-sum method [21]. Finally, in the most basic case where both $s = 1$ and $q = 1$, Block-SMOO recovers the classical stochastic gradient descent algorithm for single-objective optimization.

3 Convergence analysis

Our convergence analysis starts with the non-convex and convex cases where we show the convergence rates $\mathcal{O}(1/T^{1/2})$ for both cases. We follow up with the setting of Polyak-Łojasiewicz condition, where the convergence rate improves to $\mathcal{O}(1/T)$.

3.1 The non-convex case

We first assume that all of the objective functions are Lipschitz smooth, and the weighted-sum function F_m has a lower bound, which are standard assumptions in the optimization literature [4].

Assumption 3.1 (L-smoothness and Lower Bound) *For every $k \in \{1, \ldots, q\}$, the function $f_k(x)$ is L-smooth, i.e., there exists a constant $L > 0$ such that, $\forall x, x' \in \mathbb{R}^n$,*

$$\|\nabla f_k(x) - \nabla f_k(x')\| \leq L\|x - x'\|.$$

In addition, we assume that F_m is bounded from below, i.e., $F^* := \inf_{x \in \mathbb{R}^n} F_m(x) \in \mathbb{R}$.

For the stochastic gradients $\nabla g_k(x, \xi), k \in \{1, \ldots, q\}$, generated with random variable ξ, we use $\mathbb{E}_\xi[\cdot]$ to denote the conditional expectation taken with respect to ξ. In addition to Assumption 3.1, we impose the following two classical assumptions in stochastic approximation.

Assumption 3.2 (Unbiasedness and Bounded Gradient) *For every $k \in \{1, \ldots, q\}$, every $x \in \mathbb{R}^n$, and every realization of ξ, the stochastic gradient $\nabla g_k(x, \xi)$ satisfies the following conditions*

(a) $\mathbb{E}_\xi [\nabla g_k(x, \xi)] = \nabla f_k(x)$.

(b) $\mathbb{E}_\xi[\|\nabla g_k(x, \xi)\|^2] \leq \sigma^2$, *where σ^2 is a constant.*

The above assumptions are commonly used for analyzing gradient-type methods [5, 38, 43, 45], which guarantees that the stochastic gradient is unbiased and bounded. Note that the Block-SMOO algorithm does not need access to a stochastic estimator of the full gradient, and as a result, Assumption 3.2 only needs to hold for each partial stochastic gradient. We assume the full gradient for ease of notations.

Our main result below shows a bound for the expected squared norm gradient of the weighted-sum function F_m, which proves the sublinear convergence rate of $\mathcal{O}(1/T^{1/2})$.

Theorem 3.1 *Let Assumptions 3.1 and 3.2 hold. Let the step size be $\alpha_t = 1/\sqrt{T}$ and $\bar{x}$ be the output of Algorithm 1. We sample $\bar{x}$ uniformly at random from the set $\{x^{0,0,0}, \ldots, x^{T-1,0,0}\}$ with probability $\mathbb{P}(\bar{x} = x^{t,0,0}) = \frac{1}{T}$ for every t. We have*

$$\mathbb{E}\left[\|\nabla F_m(\bar{x})\|^2\right] \leq \frac{2}{\sqrt{T}}\left(\mathbb{E}[F_m(x^{0,0,0}) - F^*] + L\sigma^2 p^2\left[s + \frac{Ls^3}{3}\right]\right).$$

We present the proof of Theorem 3.1 in Section B of the Appendix, which we briefly summarize below. Our non-convex analysis begins by bounding the weighted-sum function $F_m(x^{t,i,j+1})$ by $F_m(x^{t,i,j})$, i.e.,

$$F_m(x^{t,i,j+1}) \leq F_m(x^{t,i,j}) + \nabla F_m(x^{t,i,j})^\top (x^{t,i,j+1} - x^{t,i,j}) + \frac{L}{2}\|x^{t,i,j+1} - x^{t,i,j}\|^2,$$

where we use the fact that F_m is also L-smooth. This observation, along with applications of Assumption 3.2 leads to the following recursive bound for the expected function value

$$\mathbb{E}[F_m(x^{t,s,p})] \leq \mathbb{E}[F_m(x^{t,0,0})] - \alpha_t \mathbb{E}\left[\sum_{i=0}^{s-1} \left\|\nabla_{\sigma(i+1)} F_m(x^{t,i,0})\right\|^2\right] + ps\alpha_t^2 \frac{L}{2}\sigma^2 + \alpha_t \mathbb{E}[A^t].$$

As the gradients are evaluated at intermediate points $x^{t,i,j}$ rather than $x^{t,i,0}$, in this bound, we introduce an error term A^t that considers these deviations,

$$\left|\sum_{i=0}^{s-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} F_m(x^{t,i,0}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right|,$$

and we show that $\mathbb{E}[A^t] \leq \alpha_t L\sigma^2 sp(p-1)$ using L-smoothness and bounded gradient assumptions.

Substituting this error bound for A^t back into the recursive bound and employing a relation between $\nabla F_m(x^{t,i,0})$ and $\nabla F_m(x^{t,0,0})$, we arrive at the following descent bound for the algorithm progress per outer iteration:

$$\mathbb{E}[F_m(x^{t+1,0,0})] \leq \mathbb{E}[F_m(x^{t,0,0})] - \frac{1}{2}\alpha_t \mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right] + \alpha_t^2 L\sigma^2 p^2\left[s + \frac{L\alpha_t s^3}{3}\right], \tag{3.1}$$

where $x^{t+1,0,0} = x^{t,s,p}$ from Algorithm 1. Applying the fixed step size $\alpha_t = \frac{1}{\sqrt{T}}$ and averaging for all t, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right] \leq \frac{2}{\sqrt{T}}\left(\mathbb{E}[F_m(x^{0,0,0}) - F^*] + L\sigma^2 p^2\left[s + \frac{Ls^3}{3}\right]\right),$$

where the final bound of Theorem 3.1 follows from the definition of $\bar{x}$. We note that this practice of choosing random output point $\bar{x}$ is standard in the literature of stochastic optimization [22].

Our convergence rate for Block-SMOO recovers the classical convergence rate $\mathcal{O}(1/T^{1/2})$ of stochastic gradient descent for single-objective optimization [22]. We note that for MOO, the work [34] does not consider the non-convex setting. Meanwhile, in single-objective optimization, [50] only considers asymptotic theoretical non-convex results for their stochastic BCD algorithm. The stochastic setting in [50] is somewhat similar to our Assumption 3.2, however, their non-convex theory assumes a vanishing variance of the stochastic gradient.

3.2 The convex case

Before presenting the convex theoretical analysis, we impose the additional assumptions as follows.

Assumption 3.3 (Convexity and Bounded Iterations) *We assume that $f_k(x)$ is convex for every $k \in \{1, \ldots, q\}$ and the solution $x^* := \arg\min_x F_m(x)$ exists. In addition, we assume that $\left\|x^{t,0,0} - x^*\right\| \leq \Delta$, for every outer iteration t of Algorithm 1.*

The iterates generated by Algorithm 1 are assumed to remain bounded. This is a standard assumption widely adopted in the literature for convex optimization and block coordinate descent methods [3, 34, 39, 50]. We note that instead of relying on this assumption, the algorithm can be modified to include a projection step onto a bounded convex set after each update. The first paragraph of our convex analysis can be easily adapted using the non-expansiveness property of orthogonal projections.

We are now ready to present our convex theorem, which bounds the expected optimality gap for the weighted-sum function F_m.

Theorem 3.2 *Let Assumptions 3.1, 3.2, and 3.3 hold. Let the step size be $\alpha_t = 1/\sqrt{T}$ and $\bar{x} = \frac{1}{T}\sum_{t=0}^{T-1} x^{t,0,0}$ be the output of Algorithm 1. We have*

$$\mathbb{E}[F_m(\bar{x}) - F^*] \leq \frac{\|x^{0,0,0} - x^*\|^2 + sp\sigma[\sigma + (\Delta L + \sigma)sp])}{2p\sqrt{T}}.$$

The proof of Theorem 3.2 is postponed to Section C of the Appendix. Our convex analysis first establishes the following recursive bound

$$\begin{aligned}\mathbb{E}[\|x^{t+1,0,0} - x^*\|^2] \leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] - 2\alpha_t p\mathbb{E}[(x^{t,0,0} - x^*)^\top(\nabla F_m(x^{t,0,0}))] \\ + \alpha_t^2 sp\sigma[\sigma + (\Delta L + \sigma)sp].\end{aligned}$$

Then we apply the fact that all objective functions are convex, therefore F_m is convex, which yields

$$\mathbb{E}[\|x^{t+1,0,0} - x^*\|^2] \leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] - 2\alpha_t p\mathbb{E}[F_m(x^{t,0,0}) - F^*] + \alpha_t^2 sp\sigma[\sigma + (\Delta L + \sigma)sp],$$

from where we obtain the final bound of Theorem 3.2 by applying a fixed step size and averaging the derived recursive bound.

One can see from the proof that the result of Theorem 3.2 only requires convexity of F_m. When the goal is to approximate the whole Pareto front, one can apply Algorithm 1 for a collection of values of the frequency vector m so that m/p discretizes sufficiently well the simplex of dimension q. In doing that, F_m has to be convex for all m, which accounts to say that all the individual functions are convex (as in Assumption 3.3).

Theorem 3.2 recovers the standard convergence rate $\mathcal{O}(1/T^{1/2})$ of BCD single-objective optimization [22, 50] and of stochastic alternating function minimization for bi-objective optimization [34].

3.3 The Polyak-Łojasiewicz case

In this section, we consider the Polyak-Łojasiewicz (PL) inequality, a generalization of strong-convexity [30, 40], which we impose for the weighted-sum function F_m.

Assumption 3.4 (Polyak-Łojasiewicz (PL) condition) *The function F_m satisfies the μ-PL inequality, i.e., there exists a constant $\mu > 0$ such that $\forall x \in \mathbb{R}^n$,*

$$\|\nabla F_m(x)\|^2 \geq 2\mu[F_m(x) - F^*],$$

where F^* is defined as in Assumption 3.1.

It is well known that a function satisfying the PL condition is not necessarily convex [30]. Under this assumption, one can show that single-objective stochastic gradient descent achieves the same theoretical rate as the sublinear rate $\mathcal{O}(1/T)$ in the strongly convex setting [13, 23]. Similarly, we show the same convergence rate for the optimality gap of Block-SMOO as follows.

Theorem 3.3 *Let Assumptions 3.1, 3.2, and 3.4 hold. Let the step size be $\alpha_t = \frac{2}{\mu(t+1)}$ and $\bar{x} = x^{T,0,0}$ be the output of Algorithm 1. We have*

$$\mathbb{E}[F_m(\bar{x}) - F^*] \leq \frac{4L\sigma^2 p^2(3\mu s + 2Ls^3)}{3T\mu^3}.$$

The proof of Theorem 3.3 follows from the bound (3.1), derived for the non-convex case.

Proof. From Assumption 3.4, one has

$$\left\|\nabla F_m(x^{t,0,0})\right\|^2 \geq 2\mu[F_m(x^{t,0,0}) - F^*].$$

Combining this bound with the inequality (3.1), for every t, we obtain

$$\mathbb{E}[F_m(x^{t+1,0,0})] \leq \mathbb{E}[F_m(x^{t,0,0})] - \frac{1}{2}\alpha_t \mathbb{E}\left[2\mu[F_m(x^{t,0,0}) - F^*]\right] + \alpha_t^2 L\sigma^2 p^2 \left[s + \frac{L\alpha_t s^3}{3}\right],$$

which is equivalent to

$$\mathbb{E}[F_m(x^{t+1,0,0}) - F^*] \leq (1 - \alpha_t\mu)\mathbb{E}[F_m(x^{t,0,0}) - F^*] + \alpha_t^2 L\sigma^2 p^2 \left[s + \frac{L\alpha_t s^3}{3}\right].$$

Choosing $\alpha_t = \frac{2}{\mu(t+1)}$ results in

$$\mathbb{E}[F_m(x^{t+1,0,0}) - F^*] \leq \frac{t-1}{t+1}\mathbb{E}[F_m(x^{t,0,0}) - F^*] + \frac{4}{\mu^2(t+1)^2}L\sigma^2p^2\left[s + \frac{2Ls^3}{3(t+1)\mu}\right],$$

which leads to

$$t(t+1)\mathbb{E}[F_m(x^{t+1,0,0}) - F^*] \leq t(t-1)\mathbb{E}[F_m(x^{t,0,0}) - F^*] + \frac{4t}{\mu^2(t+1)}L\sigma^2p^2\left[s + \frac{2Ls^3}{3\mu}\right].$$

Applying the inequality recursively for all t yields

$$\begin{aligned}(T-1)T\mathbb{E}[F_m(x^{T,0,0}) - F^*] &\leq \sum_{t=0}^{T-1}\frac{4t}{\mu^2(t+1)}L\sigma^2p^2\left[s + \frac{2Ls^3}{3\mu}\right], \\ &\leq (T-1)\frac{4}{\mu^2}L\sigma^2p^2\left[s + \frac{2Ls^3}{3\mu}\right].\end{aligned}$$

Dividing both sides by $T(T-1)$, we obtain the desired bound. □

The Polyak-Łojasiewicz convergence rate analysis of Block-SMOO matches the strongly convex convergence rate $\mathcal{O}(1/T^{1/2})$ of BCD single-objective optimization [22, 50] and the stochastic alternating function minimization for bi-objective optimization [34]. However, the Polyak-Łojasiewicz condition is weaker than assuming strong convexity, and in this case we do not need to assume bounded iterates as in [34].

4 Experiments

In this section, we illustrate the empirical performance of Block-SMOO, comparing it to other non-alternating methods.

4.1 Reduced-rank regression problem as an MOO

We consider the following multivariate reduced-rank regression problem [27, 44]:

$$\min_{U,V} \|Y - XUV\|^2, \tag{4.1}$$

where $X \in \mathbb{R}^{N\times d}$ is the feature matrix, $Y \in \mathbb{R}^{N\times q}$ is the response matrix, $U \in \mathbb{R}^{d\times r}$ and $V \in \mathbb{R}^{r\times q}$ are the low-rank factor matrices with prescribed rank r.

Multivariate regression is widely used to model multiple correlated outcomes simultaneously, e.g., in econometrics, biology, and computer science [29, 44]. Meanwhile, reduced rank regression considers the case when the variables share a low-dimensional structure, which has applications in finance, neuroimaging, and high-dimensional statistics [27, 44, 47]. As regressions for each of the q responses may convey different meanings, this problem can be recast as multi-objective optimization by treating each of the response as a separate objective, yielding

$$\min_{U,V} F(U,V) = (f_1(U,V), \ldots, f_q(U,V)), \text{ where } f_k(U,V) = \|Y_k - XUV_k\|^2, k \in \{1,\ldots,q\},$$

where $Y_k \in \mathbb{R}^N$ and $V_k \in \mathbb{R}^r$ denote the k-th column of Y and V, respectively. If one minimizes the equal-weighted-sum function $F_{1/q} = 1/q\sum_{k=1}^q f_k$ for this MOO, then the problem reduces to the reduced-rank regression problem (4.1). As our problem setting is stochastic, we compute gradient estimators from a minibatch of the dataset, which we describe below.

4.2 Datasets

We consider two datasets in this experiment. Firstly, we generate a *synthetic* dataset with $N_{\text{train}} = 2^{14}$ training samples and $N_{\text{test}} = 2^{10}$ test samples, input dimension $d = 400$, and number of responses $q = 5$. The ground-truth weight matrix $W^\star = U^\star V^\star$ has rank $r = 3$, where $U^\star \in \mathbb{R}^{d\times r}, V^\star \in \mathbb{R}^{r\times q}$ are drawn i.i.d. from $\mathcal{N}(0, I)$. Observations are generated as

$$Y = XU^\star V^\star + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{4.2}$$

with noise level $\sigma = 0.05$, and the data X are drawn i.i.d. from $\mathcal{N}(0, I)$.

In addition, we perform experiments on the *Beijing Multi-Site Air Quality* dataset [53], publicly available from the UCI Machine Learning Repository [31]. The dataset contains hourly measurements from 12 monitoring stations across Beijing from 2013 to 2017. The response matrix Y consists of the concentrations of six air pollutants: $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, and O_3. The feature matrix X comprises meteorological and temporal covariates including wind direction, wind speed, temperature, dew point, pressure, precipitation, and station and time indicators, yielding $d = 35$ features after one-hot encoding of categorical variables and removal of rows with missing values. We split the data chronologically; the first 70% of rows forms the training set and the remaining 30% forms the test set ($N_{\text{train}} \approx 265{,}000$ and $N_{\text{test}} \approx 115{,}000$). All features and responses are standardized to zero mean and unit variance using training-set statistics.

4.3 Methods and experiment setting

We evaluate our proposed Block-SMOO algorithm against three baseline methods. To ensure a fair comparison, all algorithms optimize the same equal-weighted-sum function $F_{1/q} = \frac{1}{q}\sum_{k=1}^{q} f_k$. The first baseline, denoted as the Weighted-Sum method [21], employs standard mini-batch stochastic gradient descent on $F_{1/q}$, updating all functions and coordinates simultaneously at every step. The second baseline is the Function-Alternate method (Algorithm 1 with $s = 1$), which implements stochastic alternating function minimization, a generalization of the bi-objective approach in [34]. The third baseline, termed the Block-Alternate method (Algorithm 1 with $q = 1$), applies the stochastic BCD algorithm [50] directly to the weighted-sum function. Finally, our proposed Block-SMOO method alternates the minimization of both functions and variable blocks (Algorithm 1). For Block-SMOO, we report results using a preference vector where $m_k = 2$ for all k, as this configuration empirically proved most efficient among the tested values $m_k \in \{1, 2, 4, 8\}$.

Furthermore, decision variables can be partitioned into blocks in various ways. In our experiments for both Block-SMOO and Block-Alternate, we evaluate two partitioning strategies: a two-block partition $\{U, V\}$ and a four-block partition $\{U, V^1, V^2, V^3\}$, where V^i denotes the i-th row of V. We report results from the best-performing partition for each algorithm. Additionally, for all alternating variants, the index ordering of σ and π in Algorithm 1 are randomly reshuffled at the beginning of each corresponding cycle.

All algorithms use a batch size of $B = 512$ for all datasets. The initial matrices U and V are sampled independently from $\mathcal{N}(0, 0.01I)$. Each configuration is evaluated over a grid of step sizes, i.e., $\{0.001, 0.002, 0.005, 0.01, 0.02, 0.05\}$, and the best step size for each algorithm is selected as the one achieving the lowest final test loss across the grid. We note that our method requires less computational cost per iteration, and thus for fair comparison, we run all algorithms

for a fixed budget of time $T_{\max} = 2\,\text{sec}$ for the *synthetic* dataset and $T_{\max} = 4\,\text{sec}$ for the *Beijing Multi-Site Air Quality* dataset, with test losses recorded every $\Delta t = 0.2\,\text{sec}$ and $\Delta t = 0.4\,\text{sec}$, respectively for the two datasets. Results are averaged over 10 independent random seeds.

4.4 Comparison results

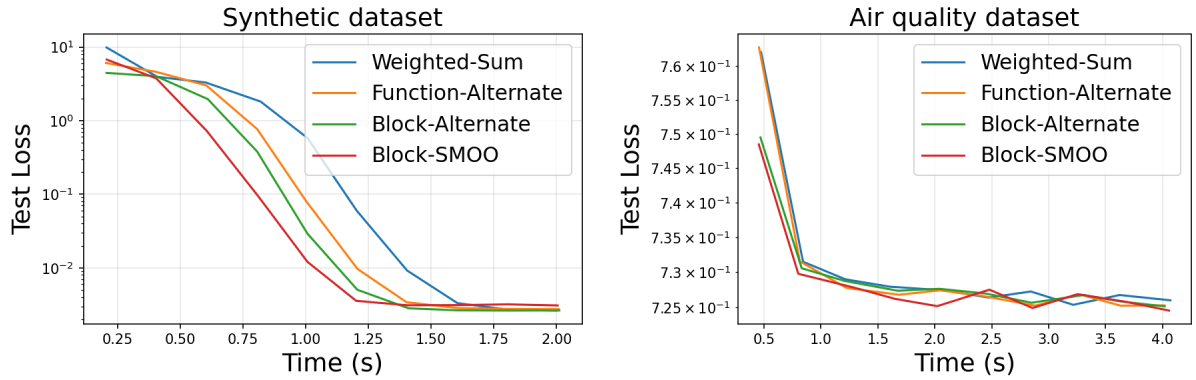

Figure 1: Test loss computed on the weighted-sum function $F_{1/q}$ versus computer time on the *synthetic* dataset (left) and the *Beijing Multi-Site Air Quality* dataset (right).

We report the test loss versus time for the datasets in Figure 1. On the *synthetic dataset*, our proposed Block-SMOO algorithm exhibits the most rapid initial convergence. It drops to a low loss significantly quicker than the standard Weighted-Sum method. While the Function-Alternate and Block-Alternate methods offer moderate improvements over the Weighted-Sum baseline, they are consistently outpaced during the initial descent by the dual-alternating approach of Block-SMOO.

On the real-world *Beijing Air Quality* dataset, Block-SMOO continues to demonstrate robust empirical performance. It consistently achieves the lowest test loss during the early stages of optimization and maintains a comparable performance over the baseline methods throughout the entire training duration. The standard Weighted-Sum method generally exhibits the slowest progress and higher overall test loss early in the run. These empirical findings support our theoretical claims and highlight the practical effectiveness gained by simultaneously alternating the update steps of objective functions and variable blocks.

4.5 Pareto front approximation

We are now interested in the ability of Block-SMOO to determine the whole Pareto front, by applying it to a collection of values of the vector m. For the *Beijing Multi-Site Air Quality* dataset above, we considered an MOO setting with only three objective functions, corresponding to the air pollutants: $PM_{2.5}$, PM_{10}, and SO_2. We compare the approximated Pareto front generated by Block-SMOO to the one from the Weighted-Sum method.

We ran Block-SMOO 231 times with $p = m_1+m_2+m_3 = 20$. Weighted-Sum was also run 231 times on the objective $F_m = 1/20(m_1 f_1 + m_2 f_2 + m_3 f_3)$, where in both cases $m_k \in \{0, \ldots, 20\}$ for all k. For both approaches, we use a fixed step size 0.02 at each iteration. Similar to the prior experiment, we sample the initial matrices U and V independently from $\mathcal{N}(0, 0.01I)$. We run

each algorithm for 20 data passes. As this Pareto front experiment is computationally expensive, we truncated the train set and the test set so that they contains $N_{\text{train}} = 2^{14}$ training samples and $N_{\text{test}} = 2^{10}$ test samples, respectively.

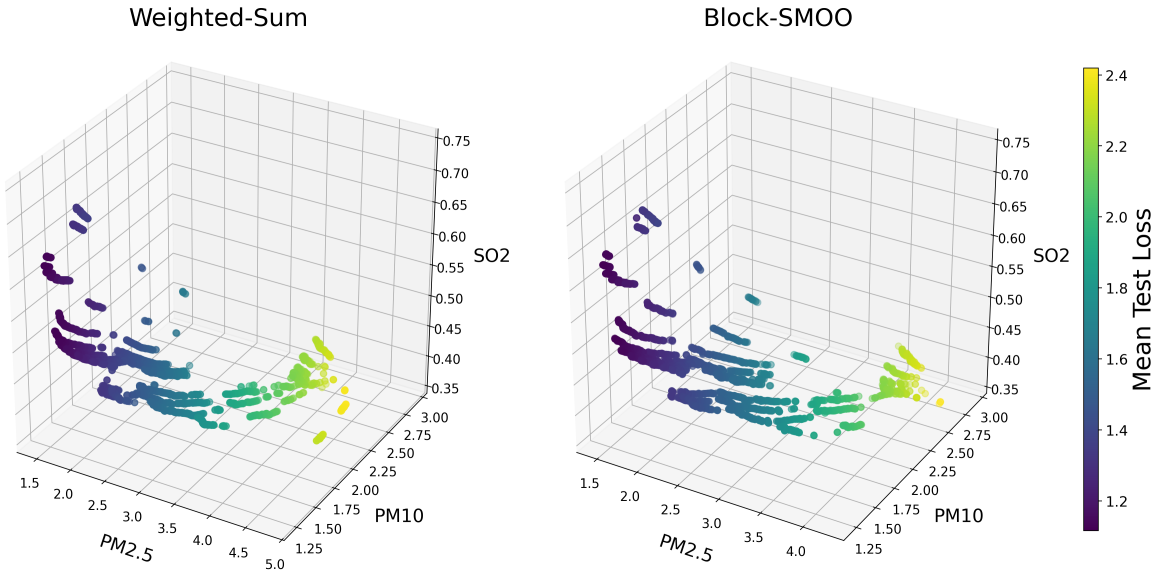

Figure 2: Approximation of Pareto fronts for test losses computed by the Weighted-Sum (left) and Block-SMOO (right) algorithms for the *Beijing Multi-Site Air Quality* dataset.

Figure 2 illustrates the approximated Pareto fronts obtained from both the Weighted-Sum and the Block-SMOO algorithms for the described problem. We computed the average test loss over three objective functions for the gradient color of each plot. Both plots demonstrate a similar structure of non-dominated solutions, suggesting that Block-SMOO effectively captures the trade-off in the Pareto front, while maintaining competitive performance across multiple objectives compared to the standard Weighted-Sum method.

Table 1: Metrics for Pareto fronts computed by the Weighted-Sum and Block-SMOO methods.

Metric	Purity	Spread Γ	Spread Δ
Weighted-Sum	0.5105	0.2878	1.1549
Block-SMOO	0.6947	0.4955	1.1785

Moreover, to assess the quality of the generated Pareto fronts, we employed two standard metrics: Purity [1] and Spread [14]. The Purity metric evaluates the accuracy of the approximation by calculating the percentage of "true" non-dominated solutions among all the non-dominated points generated by each algorithm. A higher purity ratio corresponds to a more accurate Pareto front. On the other hand, the Spread metric is designed to measure the extent of the point spread in a computed Pareto front. We computed two Spread variations: the Γ metric, which measures the maximum gap or "hole size" between points, and the Δ metric, which

indicates how well the points are distributed. Lower values for both Γ and Δ indicate a more evenly distributed and well-spread Pareto front. Detailed mathematical formula for computations of these metrics can be found in [12, 34]. We present the results in Table 1. Block-SMOO has achieved a superior accuracy with a Purity of approximately 70% (compared to 51% for Weighted-Sum), which is actually visible from Figure 2. The two methods yielded fronts with a similar value of the Δ Spread. Weighted-Sum has however achieved a better Γ spread.

5 Concluding remarks

In this paper, we demonstrated that the proposed stochastic block coordinate and function minimization algorithm (Block-SMOO) achieves convergence rates matching those of standard single-objective block coordinate descent methods. Specifically, we established sublinear convergence rates of $\mathcal{O}(1/T^{1/2})$ for both general smooth convex and non-convex functions, and an better rate of $\mathcal{O}(1/T)$ when the weighted-sum function satisfies the Polyak-Łojasiewicz (PL) condition. These theoretical guarantees are derived using applications of recursive descent bounds on the objective function value and the squared distance to optimality, established for the starting iterates in each outer loop of the algorithm.

Our current convergence analysis focuses on a two-loop structure where the outer loop iterates over the variable block indices, and the inner loop applies gradient steps to individual objective functions. Although our established bounds do not automatically hold if this two-loop structure is inverted, we anticipate that the underlying proof techniques and principles developed in this work can be adapted to that reversed setting with some minor theoretical modifications.

Other research questions arises from this work. While the current algorithm executes updates in a sequential Gauss-Seidel fashion, exploring parallel updates in a Gauss-Jacobi style represents an exciting, yet challenging direction that is particularly suitable for distributed computing problems. Furthermore, because many practical applications are characterized by non-smooth optimization landscapes, extending our framework to accommodate these cases and providing meaningful convergence guarantees for non-smooth multi-objective optimization is an important and promising topic for future research.

Acknowledgments

This work is partially supported by the U.S. Air Force Office of Scientific Research (AFOSR) award FA9550-23-1-0217 and the U.S. Office of Naval Research (ONR) award N000142412656.

Appendix

A Basic notations and lemma

In this Appendix, we present the proof of the non-convex case in Section B and the proof of the convex case in Section C. Throughout the proofs, similar to [39], we use the notation $U_i \in \mathbb{R}^{n \times n_i}$, for $i = 1, \ldots, s$, where

$$(U_1, U_2, \ldots, U_s) = I_n. \tag{A.1}$$

Then in this notation $x_i = U_i^\top x$ and $x = \sum_{i=1}^s U_i x_i$ for every $i \in \{1, \ldots, s\}$ and $x \in \mathbb{R}^n$.

Recall that in Algorithm 1, we denote by $\nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})$ the (partial) gradient of $f_{\pi(j)}$ with respect to $x_{\sigma(i+1)}$, computed at $x^{t,i,j}$. By the updates of Algorithm 1 and the definition of U_i in (A.1), we have

$$x^{t,i,j+1} = x^{t,i,j} - \alpha_t U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j}). \tag{A.2}$$

In addition, we let $\mathcal{F}_{t,i,j}$ be the σ-algebra generated by $\{x^{0,0,0}, \xi^{0,0,0}, \xi^{0,0,1}, \ldots, \xi^{t,i,j-1}\}$ i.e., $\mathcal{F}_{t,i,j}$ contains all the random information up to iteration (t, i, j). Before presenting our main analysis, we need the following Lemma.

Lemma A.1 *Let Assumption 3.2 holds. We have*

(a) For every $x \in \mathbb{R}^n$, $\|\nabla f_j(x)\| \le \sigma$ and $\|\nabla F_m(x)\| \le \sigma$.

(b) For every t, i, and j, $\mathbb{E}\left\|x^{t,i,j+1} - x^{t,i,j}\right\| \le \alpha_t \sigma$ and $\mathbb{E}\left[\left\|x^{t,i,j+1} - x^{t,i,i}\right\|^2\right] \le \alpha_t^2 \sigma^2$.

Proof.

(a) We obtain the first bound by applying the unbiased property in Assumption 3.2(a), the Jensen's inequality, and the bounded gradient condition in Assumption 3.2(b),

$$\|\nabla f_j(x)\| = \|\mathbb{E}_\xi\left[\nabla g_j(x, \xi)\right]\| \le \sqrt{\mathbb{E}_\xi\left[\|\nabla g_j(x, \xi)\|^2\right]} \le \sigma. \tag{A.3}$$

The second bound follows from the definition of F_m and the triangle inequality, i.e.,

$$\|\nabla F_m(x)\| = \left\|\frac{1}{p}\sum_{j=0}^{p-1} \nabla f_{\pi(j)}(x)\right\| \le \frac{1}{p}\sum_{j=0}^{p-1} \|\nabla f_{\pi(j)}(x)\| \overset{(A.3)}{\le} \sigma.$$

(b) From equation (A.2), we have

$$\|x^{t,i,j+1} - x^{t,i,j}\| = \alpha_t \|U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\|.$$

Applying expectation with respect to $\mathcal{F}_{t,i,j}$, then using Jensen's inequality, the fact that $\|U_{\sigma(i+1)}\| \le 1$, and Assumption 3.2(b), one obtains

$$\begin{aligned}\mathbb{E}[\|x^{t,i,j+1} - x^{t,i,j}\| | \mathcal{F}_{t,i,j}] &= \alpha_t \mathbb{E}[\|U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\| | \mathcal{F}_{t,i,j}] \\ &\le \alpha_t \sqrt{\mathbb{E}[\|U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\|^2 | \mathcal{F}_{t,i,j}]} \le \alpha_t \sigma.\end{aligned}$$

Similar arguments yield

$$\mathbb{E}[\|x^{t,i,j+1} - x^{t,i,j}\|^2 | \mathcal{F}_{t,i,j}] = \alpha_t^2 \mathbb{E}[\|U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\|^2 | \mathcal{F}_{t,i,j}] \le \alpha_t^2 \sigma^2.$$

Taking total expectation, we obtain

$$\mathbb{E}\left\|x^{t,i,j+1} - x^{t,i,j}\right\| \le \alpha_t \sigma \text{ and } \mathbb{E}\left[\left\|x^{t,i,j+1} - x^{t,i,j}\right\|^2\right] \le \alpha_t^2 \sigma^2.$$

□

B Proof of the non-convex case

We are ready to present the proof of Theorem 3.1 below.

Proof. From the definition of F_m and Assumption 3.1, F_m is also L-smooth. Based on the descent lemma [4, Proposition A.24] and Equation (A.2), we have the following bound for every t, i, j,

$$\begin{aligned}
F_m(x^{t,i,j+1}) &\leq F_m(x^{t,i,j}) + \nabla F_m(x^{t,i,j})^\top (x^{t,i,j+1} - x^{t,i,j}) + \frac{L}{2}\|x^{t,i,j+1} - x^{t,i,j}\|^2 \\
&\leq F_m(x^{t,i,j}) - \alpha_t \nabla F_m(x^{t,i,j})^\top \left(U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\right) \\
&+ \alpha_t^2 \frac{L}{2} \left\|U_{\sigma(i+1)} \nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\right\|^2 \\
&\leq F_m(x^{t,i,j}) - \alpha_t \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \left(\nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\right) \\
&+ \alpha_t^2 \frac{L}{2} \left\|\nabla_{\sigma(i+1)} g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\right\|^2,
\end{aligned}$$

where the last equation used $\|U_{\sigma(i+1)}\|^2 \leq 1$ and $g_i^\top = g^\top U_i$. Applying expectation with respect to $\mathcal{F}_{t,i,j}$ to the previous bound followed by Assumption 3.2, one obtains

$$\mathbb{E}[F_m(x^{t,i,j+1})|\mathcal{F}_{t,i,j}] \leq F_m(x^{t,i,j}) - \alpha_t \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) + \alpha_t^2 \frac{L}{2}\sigma^2.$$

Taking total expectation on both sides of this bound and applying the inequality recursively for all i, j, we have

$$\begin{aligned}
\mathbb{E}[F_m(x^{t,s,p})] &\leq \mathbb{E}[F_m(x^{t,0,0})] - \alpha_t \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \mathbb{E}\left[\nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right] + ps\alpha_t^2 \frac{L}{2}\sigma^2 \\
&\leq \mathbb{E}[F_m(x^{t,0,0})] - \alpha_t \mathbb{E}\left[\sum_{i=0}^{s-1} \left\|\nabla_{\sigma(i+1)} F_m(x^{t,i,0})\right\|^2\right] + ps\alpha_t^2 \frac{L}{2}\sigma^2 + \alpha_t \mathbb{E}[A^t], \qquad \text{(B.1)}
\end{aligned}$$

where A^t is the following term:

$$\begin{aligned}
&\left|\sum_{i=0}^{s-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} F_m(x^{t,i,0}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right| \\
&= \left|\sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,0}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right| \\
&\leq \left|\sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,0}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right| \\
&+ \left|\sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,0})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \nabla_{\sigma(i+1)} F_m(x^{t,i,j})^\top \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right| \\
&\leq \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \left\|\nabla_{\sigma(i+1)} F_m(x^{t,i,0})\right\| \cdot \left\|\nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,0}) - \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j})\right\|
\end{aligned}$$

$$+\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\left\|\nabla_{\sigma(i+1)}F_m(x^{t,i,0})-\nabla_{\sigma(i+1)}F_m(x^{t,i,j})\right\|\cdot\left\|\nabla_{\sigma(i+1)}f_{\pi(j)}(x^{t,i,j})\right\|$$
$$\leq\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\sigma L\left\|x^{t,i,0}-x^{t,i,j}\right\|+\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\sigma L\left\|x^{t,i,0}-x^{t,i,j}\right\|=2\sigma L\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\left\|x^{t,i,0}-x^{t,i,j}\right\|,$$

where we used the L-smoothness and bounded gradient of F_m and $f_{\pi(j)}$. Taking expectation of both sides, we arrive at

$$\begin{aligned}\mathbb{E}[A^t]&\leq 2\sigma L\mathbb{E}\left[\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\left\|x^{t,i,0}-x^{t,i,j}\right\|\right]=2\sigma L\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\mathbb{E}\left[\left\|x^{t,i,0}-x^{t,i,j}\right\|\right]\\&\leq 2\sigma L\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\mathbb{E}\left[\sum_{k=0}^{j-1}\left\|x^{t,i,k+1}-x^{t,i,k}\right\|\right]\leq 2\sigma L\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}j\alpha_t\sigma\leq\alpha_t L\sigma^2 sp(p-1),\end{aligned}$$

where the last line follows from the triangle inequality and Lemma A.1(b).

Substituting $\mathbb{E}[A^t]$ into our main derivation (B.1) yields

$$\begin{aligned}\mathbb{E}[F_m(x^{t,s,p})]&\leq\mathbb{E}[F_m(x^{t,0,0})]-\alpha_t\mathbb{E}\left[\sum_{i=0}^{s-1}\left\|\nabla_{\sigma(i+1)}F_m(x^{t,i,0})\right\|^2\right]+ps\alpha_t^2\frac{L}{2}\sigma^2+\alpha_t^2L\sigma^2sp(p-1)\\&\leq\mathbb{E}[F_m(x^{t,0,0})]-\frac{1}{2}\alpha_t\mathbb{E}\left[\sum_{i=0}^{s-1}\left\|\nabla_{\sigma(i+1)}F_m(x^{t,0,0})\right\|^2\right]\\&\quad+\alpha_t\mathbb{E}\left[\sum_{i=0}^{s-1}\left\|\nabla_{\sigma(i+1)}F_m(x^{t,i,0})-\nabla_{\sigma(i+1)}F_m(x^{t,0,0})\right\|^2\right]+\alpha_t^2L\sigma^2s\left[p(p-1)+\frac{p}{2}\right],\end{aligned}$$

where we used the inequality $-a^2\leq-\frac{1}{2}b^2+(a-b)^2$. Using the L-smooth property of F_m and recalling from Algorithm 1 that $x^{t+1,0,0}=x^{t,s,p}$, one obtains

$$\begin{aligned}\mathbb{E}[F_m(x^{t+1,0,0})]&\leq\mathbb{E}[F_m(x^{t,0,0})]-\frac{1}{2}\alpha_t\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right]\\&\quad+\alpha_tL^2\mathbb{E}\left[\sum_{i=0}^{s-1}\left\|x^{t,i,0}-x^{t,0,0}\right\|^2\right]+\alpha_t^2L\sigma^2sp^2\\&\leq\mathbb{E}[F_m(x^{t,0,0})]-\frac{1}{2}\alpha_t\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right]\\&\quad+\alpha_tL^2\sum_{i=0}^{s-1}\mathbb{E}\left[ip\sum_{k=0}^{i-1}\sum_{j=0}^{p-1}\left\|x^{t,k,j+1}-x^{t,k,j}\right\|^2\right]+\alpha_t^2L\sigma^2sp^2\\&\leq\mathbb{E}[F_m(x^{t,0,0})]-\frac{1}{2}\alpha_t\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right]+\alpha_tL^2\sum_{i=0}^{s-1}i^2p^2\alpha_t^2\sigma^2+\alpha_t^2L\sigma^2p^2\\&\leq\mathbb{E}[F_m(x^{t,0,0})]-\frac{1}{2}\alpha_t\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right]+\alpha_t^2L\sigma^2p^2\left[s+\frac{L\alpha_ts^3}{3}\right],\end{aligned}$$

where the second bound follows from the inequality $\|\sum_{i=1}^n a_i\|^2 \leq n\sum_{i=1}^n \|a_i\|^2$ and the third bound follows from Lemma A.1(b). Rearranging the derived bound gives us

$$\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right] \leq \frac{2}{\alpha_t}\left(\mathbb{E}[F_m(x^{t,0,0})] - \mathbb{E}[F_m(x^{t+1,0,0})]\right) + 2\alpha_t L\sigma^2 p^2\left[s + \frac{L\alpha_t s^3}{3}\right].$$

Averaging for all t, one obtains

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right] \leq \frac{1}{T}\sum_{t=0}^{T-1}\frac{2}{\alpha_t}\left(\mathbb{E}[F_m(x^{t,0,0})] - \mathbb{E}[F_m(x^{t+1,0,0})]\right)\\ + \frac{1}{T}\sum_{t=0}^{T-1} 2\alpha_t L\sigma^2 p^2\left[s + \frac{L\alpha_t s^3}{3}\right].\end{aligned}$$

Setting $\alpha_t = \frac{1}{\sqrt{T}}$ yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F_m(x^{t,0,0})\right\|^2\right] \leq \frac{2}{\sqrt{T}}\left(\mathbb{E}[F_m(x^{0,0,0}) - F^*] + L\sigma^2 p^2\left[s + \frac{Ls^3}{3}\right]\right),$$

which concludes the proof. □

C Proof of the convex case

We present the proof of Theorem 3.2 as follows.

Proof. For every iteration t, i, j, applying Equation (A.2) yields

$$\begin{aligned}\|x^{t,i,j+1} - x^*\|^2 &= \|(x^{t,i,j} - x^*) + (x^{t,i,j+1} - x^{t,i,j})\|^2\\ &= \|x^{t,i,j} - x^*\|^2 + 2(x^{t,i,j} - x^*)^\top(x^{t,i,j+1} - x^{t,i,j}) + \|x^{t,i,j+1} - x^{t,i,j}\|^2\\ &= \|x^{t,i,j} - x^*\|^2 - 2\alpha_t(x^{t,i,j} - x^*)^\top(U_{\sigma(i+1)}\nabla_{\sigma(i+1)}g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j}))\\ &+ \alpha_t^2\|U_{\sigma(i+1)}\nabla_{\sigma(i+1)}g_{\pi(j)}(x^{t,i,j}, \xi^{t,i,j})\|^2.\end{aligned}$$

Applying expectation with respect to $\mathcal{F}_{t,i,j}$ on both sides of the previous bound and using Assumption 3.2, we obtain

$$\mathbb{E}[\|x^{t,i,j+1} - x^*\|^2|\mathcal{F}_{t,i,j}] \leq \|x^{t,i,j} - x^*\|^2 - 2\alpha_t(x^{t,i,j} - x^*)^\top(U_{\sigma(i+1)}\nabla_{\sigma(i+1)}f_{\pi(j)}(x^{t,i,j})) + \alpha_t^2\sigma^2.$$

Taking total expectation of the previous bound and applying the inequality recursively for all i, j, one has

$$\begin{aligned}&\mathbb{E}[\|x^{t,s,p} - x^*\|^2]\\ &\leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] - 2\alpha_t\sum_{i=0}^{s-1}\sum_{j=0}^{p-1}\mathbb{E}[(x^{t,i,j} - x^*)^\top U_{\sigma(i+1)}\nabla_{\sigma(i+1)}f_{\pi(j)}(x^{t,i,j})] + sp\alpha_t^2\sigma^2\\ &\leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] - 2\alpha_t p\mathbb{E}[(x^{t,0,0} - x^*)^\top(\nabla F_m(x^{t,0,0}))] + sp\alpha_t^2\sigma^2 + 2\alpha_t\mathbb{E}[B^t], \qquad \text{(C.1)}\end{aligned}$$

where B^t is the following term:

$$\left| p(x^{t,0,0} - x^*)^\top (\nabla F_m(x^{t,0,0})) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,i,j} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) \right|.$$

Let us analyze the first term in B^t using the definitions of U_i and F_m, writing

$$\begin{aligned}
&p(x^{t,0,0} - x^*)^\top (\nabla F_m(x^{t,0,0})) = (x^{t,0,0} - x^*)^\top \left(p \sum_{i=0}^{s-1} U_{\sigma(i+1)} \nabla_{\sigma(i+1)} F_m(x^{t,0,0}) \right) \\
&= (x^{t,0,0} - x^*)^\top \left(\sum_{i=0}^{s-1}\sum_{j=0}^{p-1} U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,0,0}) \right) \\
&= \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,0,0} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,0,0}).
\end{aligned}$$

As a result, B^t is upper bounded by

$$\begin{aligned}
&\left| \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,0,0} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,0,0}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,0,0} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) \right| \\
&+ \left| \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,0,0} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) - \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} (x^{t,i,j} - x^*)^\top U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) \right| \\
&\leq \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \left\| x^{t,0,0} - x^* \right\| \cdot \left\| U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,0,0}) - U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) \right\| \\
&+ \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \left\| x^{t,0,0} - x^{t,i,j} \right\| \cdot \left\| U_{\sigma(i+1)} \nabla_{\sigma(i+1)} f_{\pi(j)}(x^{t,i,j}) \right\| \\
&\leq \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \Delta \cdot L \left\| x^{t,0,0} - x^{t,i,j} \right\| + \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \left\| x^{t,0,0} - x^{t,i,j} \right\| \cdot \sigma,
\end{aligned}$$

where the last line follows from the L-smoothness and bounded gradient properties of $f_{\pi(j)}$, and the fact that $\left\| x^{t,0,0} - x^* \right\| \leq \Delta$. Applying expectation on both sides of the previous bound, one obtains

$$\begin{aligned}
&\mathbb{E}[B^t] \leq (\Delta L + \sigma) \mathbb{E} \left[\sum_{i=0}^{s-1}\sum_{j=0}^{p-1} \left\| x^{t,0,0} - x^{t,i,j} \right\| \right] \\
&\leq (\Delta L + \sigma) \sum_{i=0}^{s-1}\sum_{j=0}^{p-1} [(ip + j)\alpha_t \sigma] = \frac{1}{2} \alpha_t \sigma (\Delta L + \sigma) \left(s(s-1)p^2 + p^2 s \right) = \frac{1}{2} \alpha_t \sigma (\Delta L + \sigma) s^2 p^2,
\end{aligned}$$

where the last line follows from the triangle inequality and Lemma A.1(b).

Substituting $\mathbb{E}[B^t]$ into our main derivation (C.1) give us

$$\mathbb{E}[\|x^{t,s,p} - x^*\|^2] \leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] \\ - 2\alpha_t p \mathbb{E}[(x^{t,0,0} - x^*)^\top (\nabla F_m(x^{t,0,0}))] + \alpha_t^2 sp\sigma[\sigma + (\Delta L + \sigma)sp].$$

As f_j are convex for $j \in \{1, \ldots, q\}$, F_m is convex, and we have

$$\nabla F_m(x^{t,0,0})^\top \left(x^{t,0,0} - x^*\right) \geq F_m(x^{t,0,0}) - F^*.$$

Applying this bound and $x^{t+1,0,0} = x^{t,s,p}$ yield

$$\mathbb{E}[\|x^{t+1,0,0} - x^*\|^2] \leq \mathbb{E}[\|x^{t,0,0} - x^*\|^2] - 2\alpha_t p \mathbb{E}[F_m(x^{t,0,0}) - F^*] + \alpha_t^2 sp\sigma[\sigma + (\Delta L + \sigma)sp].$$

Using the last inequality recursively for all t and rearranging, we obtain

$$\sum_{t=0}^{T-1} 2\alpha_t p \mathbb{E}[F_m(x^{t,0,0}) - F^*] \leq \|x^{0,0,0} - x^*\|^2 - \mathbb{E}[\|x^{T,0,0} - x^*\|^2] + \sum_{t=0}^{T-1} \alpha_t^2 sp\sigma[\sigma + (\Delta L + \sigma)sp].$$

Choosing $\alpha_t = \frac{1}{\sqrt{T}}$ results in

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[F_m(x^{t,0,0}) - F^*] \leq \frac{\|x^{0,0,0} - x^*\|^2 + sp\sigma[\sigma + (\Delta L + \sigma)sp]}{2p\sqrt{T}}.$$

As F_m is convex and $F_m(\bar{x}) \leq \frac{1}{T} \sum_{t=0}^{T-1} F_m(x^{t,0,0})$, we have the desired bound.

□

References

[1] 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.

[2] S. Bechikh, R. Datta, and A. Gupta, editors. *Recent Advances in Evolutionary Multi-Objective Optimization*, volume 20. Springer, New York, 2016.

[3] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM J. Optim.*, 23:2037–2060, 2013.

[4] D. P. Bertsekas. Nonlinear programming. *J. Oper. Res. Soc.*, 48:334–334, 1997.

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

[6] E. K. Browning and M. A. Zupan. *Microeconomics: Theory and Applications.* Wiley, Hoboken, 2020.

[7] X. Cai, C. Song, S. Wright, and J. Diakonikolas. Cyclic block coordinate descent with variance reduction for composite nonconvex optimization. *International Conference on Machine Learning*, 202:3469–3494, 2023.

[8] L. Chen, H. Fernando, Y. Ying, and T. Chen. Three-Way Trade-Off in Multi-Objective Learning: Optimization, Generalization and Conflict-Avoidance. *Advances in Neural Information Processing Systems*, 36:70045–70093, 2023.

[9] P. Cheng, J. Sulaiman, K. Ghazali, M. K. M. Ali, and M. Xu. Application of Newton-Gauss-Seidel method for solving multi-objective constrained optimization problems. *Transactions on Science and Technology*, 11:43–50, 2024.

[10] C. C. Coello. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1:28–36, 2006.

[11] K. H. Cuevas. *Cyclic stochastic optimization: Generalizations, convergence, and applications in multi-agent systems.* PhD thesis, The Johns Hopkins University, 2017.

[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] S. De, A. Yadav, D. Jacobs, and T. Goldstein. Automated Inference with Adaptive Batches. *International Conference on Artificial Intelligence and Statistics*, 54:1504–1513, 2017.

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

[16] D. Driggs, J. Tang, J. Liang, M. Davies, and C. Schonlieb. A stochastic proximal alternating minimization for nonsmooth and nonconvex optimization. *SIAM J. Imaging Sci.*, 14:1932–1970, 2021.

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

[18] H. Fernando, H. Shen, M. Liu, S. Chaudhury, K. Murugesan, and T. Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. *International Conference on Learning Representation*, 2023.

[19] H. Fernando, L. Chen, S. Lu, P. Chen, M. Liu, S. Chaudhury, K. Murugesan, G. Liu, M. Wang, and T. Chen. Variance reduction can improve trade-off in multi-objective learning. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6975–6979, 2024.

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

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

[22] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23:2341–2368, 2013.

[23] R. Gower, O. Sebbouh, and N. Loizou. SGD for structured nonconvex functions: Learning rates, minibatching and interpolation. *International Conference on Artificial Intelligence and Statistics*, 130:1315–1323, 2021.

[24] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Oper. Res. Lett.*, 26:127–136, 2000.

[25] N. Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5:1502242, 2018.

[26] 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.

[27] A. J. Izenman. Reduced-rank regression for the multivariate linear model. *J. Multivariate Anal.*, 5:248–264, 1975.

[28] T. Jiang and L. Xiao. Stochastic approximation with block coordinate optimal stepsizes. *arXiv preprint arXiv:2507.08963*, 2025.

[29] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, Saddle River, 2002.

[30] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. *Machine Learning and Knowledge Discovery in Databases*, 9851:795–811, 2016.

[31] M. Kelly, R. Longjohn, and K. Nottingham. The UCI Machine Learning Repository, 2024.

[32] J. Liu, S. Wright, C. Re, V. Bittorf, and S. Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *International Conference on Machine Learning*, 32:469–477, 2014.

[33] S. Liu and L. N. Vicente. Convergence rates of the stochastic alternating algorithm for bi-objective optimization. *J. Optim. Theory Appl.*, 198:165–186, 2023.

[34] S. Liu and L. N. Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Ann. Oper. Res.*, 339: 1119–1148, 2024.

[35] Z. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.*, 72:7–35, 1992.

[36] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.*, 26:369–395, 2004.

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

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

[39] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22:341–362, 2012.

[40] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:1–17, 1964.

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

[42] M. Razaviyayn, M. Hong, and Z. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM J. Optim.*, 23:1126–1153, 2013.

[43] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*, 24:693–701, 2011.

[44] G. C. Reinsel and R. P. Velu. *Multivariate Reduced-rank Regression.* Springer, New York, 1998.

[45] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. *International Conference on Machine Learning*, pages 807–814, 2007.

[46] H. M. Shi, S. Tu, Y. Xu, and W. Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016.

[47] M. Vounou, T. E. Nichols, G. Montana, and Alzheimer's Disease Neuroimaging Initiative. Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. *Neuroimage*, 53:1147–1159, 2010.

[48] Z. Wen, D. Goldfarb, and K. Scheinberg. Block coordinate descent methods for semidefinite programming. *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 533–564, 2012.

[49] S. J. Wright. Coordinate descent algorithms. *Math. Program.*, 151:3–34, 2015.

[50] Y. Xu and W. Yin. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM J. Optim.*, 25:1686–1716, 2015.

[51] Y. Yang, M. Pesavento, Z. Luo, and B. Ottersten. Inexact block coordinate descent algorithms for nonsmooth nonconvex optimization. *IEEE Transactions on Signal Processing*, 68:947–961, 2019.

[52] Z. Yu and D. W. Ho. Zeroth-order stochastic block coordinate type methods for nonconvex optimization. *arXiv preprint arXiv:1906.05527*, 2019.

[53] S. Zhang. Beijing Multi-Site Air-Quality Data. UCI Machine Learning Repository, 2019.

[54] S. Zhou, W. Zhang, J. Jiang, W. Zhong, J. Gu, and W. Zhu. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115, 2022.