On the Convergence of Stochastic Multi-Objective Gradient Manipulation and Beyond

Shiji Zhou*
Tsinghua University
zsj17@mails.tsinghua.edu.cn

Wenpeng Zhang†
Ant Group
zhangwenpeng0@gmail.com

Jiyan Jiang
Tsinghua University
scjjy95@outlook.com

Wenliang Zhong
Ant Group
lihong.glh@antgroup.com

Jinjie Gu
Ant Group
lihong.glh@antgroup.com

Wenwu Zhu†
Tsinghua University
wwzhu@tsinghua.edu.cn

Abstract

The conflicting gradients problem is one of the major bottlenecks for the effective training of machine learning models that deal with multiple objectives. To resolve this problem, various gradient manipulation techniques, such as PCGrad, MGDA, and CAGrad, are developed, which directly manipulate the conflicting gradients to refined ones with alleviated or even no conflicts. However, the existing design and analysis of these techniques have been mainly conducted under the exact full-batch gradient setting, ignoring the fact that they are primarily applied with stochastic mini-batch gradients. In this paper, we point out that the stochastic gradient manipulation algorithms may fail to converge to Pareto optimal solutions. Firstly, we show that these seemingly different algorithms can be summarized into a unified algorithmic framework, where the descent direction is given by the composition of the gradients w.r.t. the multiple objectives. Then we provide a simple two-objective convex optimization instance to illustrate the non-convergence issue in detail under the unified framework, where the directions yielded by PCGrad, MGDA, and CAGrad even reversely optimize all the objectives in expectation simultaneously. The analysis also shows that the non-convergence results from the determination of the composite weights solely by the instantaneous stochastic gradients. To fix the non-convergence issue, we propose a novel composite weights determination scheme that exponentially averages the past calculated weights. Finally, we show the resulting new variants of stochastic gradient manipulation converge to Pareto optimality and give rise to comparable or improved empirical performance.

1 Introduction

Multi-objective optimization (MOO) refers to the optimization of multiple objectives simultaneously [10, 6, 5]. MOO problems appear in many real-world scenarios, e.g., the online advertising model [35, 36, 29, 45] aiming to maximize both the Click-Through Rate (CTR) and the Post-Click Conversion Rate (CVR); autonomous driving system [16, 34] learning to solve multiple tasks such as

*This work was supported by Ant Group through Ant Research Intern Program
†Corresponding Author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

self-localization and object identification at the same time. The main challenge lies in the inherent conflicts between multiple objectives that would impair the training [21, 48, 30]. For example, in online advertising, merely optimizing CTR or CVR will degrade the performance of the other [35, 36]. Various MOO algorithms [40, 48, 30, 44, 31, 18] were proposed to deal with the conflicting problem by altering the gradients evaluated from multi-objective losses. Typically, multi-objective gradient manipulation (MOGM) methods such as MGDA [40], PCGrad [48], and CAGrad [30] adjust the multiple gradients to seek a common descent direction that simultaneously descents all the objectives.

While these algorithms have been successfully applied in a wide range of applications and achieved significant performance gains, the current design and analysis are only derived with exact (i.e., full-batch) objective gradients [40, 48, 30, 44, 31, 18]. However, in most real-world scenarios, evaluating stochastic (i.e., mini-batch) gradients is a much more practical and efficient method from the computational cost aspect, especially for training deep neural networks (DNNs) [24, 38].

In the single-objective case, since the stochastic gradients are unbiased estimators of the true gradients, directly using the stochastic gradient still maintains the right expected direction to descend the objective. However, in the multi-objective case, since altering the stochastic gradients could amplify the negative impact of the stochastic noise, the common descent direction for the stochastic objectives may not be the right direction to descend the true objectives in expectation. Such gap between full and stochastic gradients could cause non-convergence issues for directly using MOGM algorithms.

In this paper, we rigorously show that the above non-convergence issues indeed exist. Surprisingly, we prove that *altering the stochastic gradients yields a direction that even inversely optimizes all the objectives in expectation* for some cases. Such an inverse direction that moves the algorithms in the wrong direction will resist the solution from convergence. This implies that current MOGM algorithms are suboptimal under stochastic gradient settings, which potentially degrades the performance of MOGM algorithms applied for training DNNs. In particular, we make the following key contributions.

To uniformly analyze different MOGM algorithms, we summarize them into a general framework where the altered direction is a weighted composition of the multiple gradients, and the composite weights are functions of the multiple gradients determined by the alteration rules of different MOGM algorithms. We provide strictly equivalent transformation to show that typical methods such as MGDA, PCGrad, and CAGrad are all included in the summarized form. This framework is general and incorporates a wide range of algorithms, such as recently proposed GradVac [44], IMTL-G [31].

Under the above general framework, we elucidate how the MOGM algorithms can cause non-convergence by providing an example of a simple stochastic convex optimization problem, where MGDA, PCGrad, and CAGrad provably do not converge to an optimal solution. The key reason is that the composite weights are calculated only from the instantaneous gradients in one step, and thus inherit the same randomness as the stochastic gradients. This strong correlation between the composite weights and stochastic gradients could lead to significant estimation bias of their composition, which forces the composite gradient in the wrong direction that even degrades all the objectives. Since MGDA, PCGrad, and CAGrad possess typical properties among MOGM algorithms, our analysis easily extends to other MOGM algorithms contained in our general framework as well. Furthermore, we point out that the previous analysis of the convergence for the MGDA algorithm [33] is established on an unjustified assumption, where the composite weights are Lipschitz functions w.r.t. each gradient. We rigorously prove that this assumption is not true for MOGM algorithms.

The above analysis suggests that in order to achieve guaranteed convergence, the optimization algorithms must reduce the correlation between the composite weights and stochastic gradients. To resolve this issue while preserving the practical advantages of MOGM, we fix MOGM algorithms with a principled and practical momentum mechanism that exponentially averages the composite weights instead of just using the values computed from the instantaneous gradients. Theoretically, momentum reduces the variance of the composite weight [4], then decreasing the correlation as well. Thus, the inverse direction problem can be alleviated, and we prove that gradient manipulation algorithms can converge to optimal with the same rate as single objective stochastic optimization by adopting this mechanism. Our approach is simple to implement and agnostic to model and optimizer, and hence can be easily plugged into various gradient manipulation algorithms.

Finally, we empirically verify our proposal in simulation and deep multi-task learning tasks, where we observe comparable or better performance. Our advantage becomes more significant when the batch size is smaller, implying that the convergence issue is more severe with larger stochastic noise.

2 Preliminaries

In this section, we briefly review the basic background knowledge of multiple-objective optimization, and introduce typical algorithms as well as their convergence analysis.

2.1 Basic Concepts of Multiple-Objective Optimization

Multiple-objective optimization (MOO) is concerned with solving the problems of optimizing multiple objective functions simultaneously [50, 40], termed as

$$\min_{\boldsymbol{x}\in\mathcal{K}} \boldsymbol{F}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))^\top, \tag{1}$$

where $m \geq 2$ denotes the number of objectives, and $\mathcal{K} \subset \mathbb{R}^d$ denotes the feasible set in a d dimension space. Each objective $f^i : \mathcal{K} \to \mathbb{R}$, $i \in \{1, \ldots, m\}$ is the i-th loss function. Different from single-objective optimization where solutions $\boldsymbol{x}, \boldsymbol{x}'$ can be order by $f(\boldsymbol{x}) \leq f(\boldsymbol{x}')$ or $f(\boldsymbol{x}) \geq f(\boldsymbol{x}')$. In MOO, it could have two parameter vectors where one performs better for task i and the other performs better for task $j \neq i$. Therefore, Pareto optimality is defined to deal with such an incomparable case.

Definition 1 (**Pareto optimality**). *For any two solutions $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{K}$, we say that $\boldsymbol{x}$ dominates $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \prec \boldsymbol{x}'$ or $\boldsymbol{x}' \succ \boldsymbol{x}$, if $f^i(\boldsymbol{x}) \leq f^i(\boldsymbol{x}')$ for all i, and there exists one i such that $f^i(\boldsymbol{x}) < f^i(\boldsymbol{x}')$; otherwise, we say that $\boldsymbol{x}$ does not dominate $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \not\prec \boldsymbol{x}'$ or $\boldsymbol{x}' \not\succ \boldsymbol{x}$. A solution $\boldsymbol{x}^* \in \mathcal{K}$ is called Pareto optimal if it is not dominated by any other solution in $\mathcal{K}$.*

Note that there is a set of Pareto optimal solutions, termed as **Pareto set**. The goal of MOO is to find a Pareto optimal solution, which is necessary to be Pareto critical [3].

Definition 2 (**Pareto criticality**). *A solution $\boldsymbol{x}^* \in \mathcal{K}$ is called Pareto critical if there is no common descent direction $\boldsymbol{d} \in \mathbb{R}^d$ such that $\nabla f^i(\boldsymbol{x}^*)^\top \boldsymbol{d} < 0, i = 1, \ldots m$ for all objectives.*

This definition indicates that if $\boldsymbol{x}$ is not Pareto critical, such direction $\boldsymbol{d}$ will be a local descent direction for $\boldsymbol{F}$ at point $\boldsymbol{x}$. Optimizing through $\boldsymbol{d}$ in the local neighborhood of $\boldsymbol{x}$ is able to get a better solution that dominates $\boldsymbol{x}$ [10]. Since Pareto criticality reflects the local property compared with Pareto optimality, it is often used as the local minimal condition for MOO with non-convex objectives [11]. We then present sufficient conditions for determining Pareto criticality/optimality, which appear as metrics to study the convergence for MOO algorithm [11, 43]. Denotes the probability simplex as $S_m = \{(\lambda^1, \ldots, \lambda^m) | \sum_{i=1}^m \lambda^i = 1, \lambda \in [0, 1]\}$, we have the following proposition.

Proposition 1. *For MOO problem 1, **(a)** If there exists $\boldsymbol{\lambda} \in S_m$ such that $\| \sum_{i=1}^m \lambda^i \nabla f^i(\boldsymbol{x}^*) \| = 0$, then $\boldsymbol{x}^* \in \mathcal{K}$ is Pareto critical. **(b)** For convex objectives $f^i(\cdot), i = 1, \ldots, m$. If there exists $\boldsymbol{\lambda} \in S_m$ such that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{K}} \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x})$, then $\boldsymbol{x}^*$ is Pareto optimal*[3].

(a) is derived from the equivalent definition of Pareto critical points in [30], it reflects the stationary condition for MOO algorithms. (b) refers to Theorem 5.13 and Lemma 5.14 in [17], it implies that the minimizer of any linearization is Pareto optimal for the multiple objectives.

2.2 Multi-Objective Gradient Manipulation

Similar to single-objective optimization, MOO can be solved by running iteratively with gradient-based algorithms. Suppose n denotes the number of iterations. At each $k = 1, \ldots, n$ round, the key challenge is the gradient conflicting problem, i.e., $\nabla f^i(\boldsymbol{x}_k)^\top \nabla f^j(\boldsymbol{x}_k) < 0, i \neq j, i, j \in [m] = \{1, \ldots, m\}$, which leads to performance drop for an objective when optimizing another one [48], i.e. optimizing f^i along the negative gradient direction $-\nabla f^i(\boldsymbol{x}_k)$ will inversely optimize f^j. multi-objective gradient manipulation (MOGM) algorithms aim to search for a direction $\boldsymbol{d}_k$ that is not conflicting with each negative gradient, i.e., $-\nabla f^i(\boldsymbol{x}_k)\boldsymbol{d}_k > 0, i \in [m]$. Using such a non-conflicting direction $\boldsymbol{d}_k$ to execute the gradient descent step to update the decision i.e., $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta \boldsymbol{d}_k$ where η is the step size, is shown to get better performance in practice [40, 48, 30]. Mathematically, we can measure the decrease of each objective $i \in [m]$

$$f^i(\boldsymbol{x}_k) - f^i(\boldsymbol{x}_k + \eta \boldsymbol{d}_k) \approx -\eta \boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k)$$

[3] Precisely, $\boldsymbol{x}^*$ is weak Pareto optimal. If $\boldsymbol{x}^*$ is the unique minimizer, $\boldsymbol{x}^*$ is Pareto optimal. For convenience of understanding, we do not distinguish them in the main text, and provide detailed comparison in Appendix A

by the first-order Taylor approximation assuming η is small [30]. Therefore, if $-\boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k) > 0$ for each objective i, then this direction is able to descend all objectives simultaneously. There are various ways to compute such kind of direction, and we'll introduce typical MOGM methods as follows.

Multiple Gradient Descent Algorithm (MGDA) [40] directly optimizes towards the Pareto criticality by leveraging Definition 2. Specifically, in each iteration k, MGDA aims to find a direction $\boldsymbol{d}_k$ to maximize the minimum decrease across the losses by solving the following subproblem

$$\max_{\boldsymbol{d}\in\mathbb{R}^d} \min_{i\in[m]} (f^i(\boldsymbol{x}_k) - f^i(\boldsymbol{x}_k + \eta \boldsymbol{d}_k)) \approx -\eta \min_{\boldsymbol{d}\in\mathbb{R}^d} \max_{i\in[m]} \nabla f^i(\boldsymbol{x}_k)^\top \boldsymbol{d}_k.$$

By regularizing the norm of $\boldsymbol{d}_k$ on the right side, it computes the direction by

$$\boldsymbol{d}_k = \arg\min_{\boldsymbol{d}\in\mathbb{R}^d} \{\max_{i\in[m]} \nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d} + \frac{1}{2}\|\boldsymbol{d}\|_2^2\}.$$

This sub-problem can be rewritten equivalently as the following differentiable quadratic optimization

$$\boldsymbol{d}_k, \mu = \arg\min_{\{d\in\mathbb{R}^d, \mu\in\mathbb{R}\}} (\frac{1}{2}\|\boldsymbol{d}\|_2^2 + \mu), \text{ s.t. } \nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d} \leq \mu.$$

If $\mu < 0$, then $\nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d}_k < 0$, which means $\boldsymbol{x}_k$ is not Pareto critical from Definition 2, and $\boldsymbol{d}_k$ is the direction to descent all the objectives simultaneously [10, 11]. To simplify the optimization, such primal problem has a dual objective as a min-norm oracle

$$\boldsymbol{\lambda}_k = \arg\min_{\lambda_k\in S_m} \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|.$$

The direction is then calculated by $\boldsymbol{d}_k = -\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)$.

Projecting Conflicting Gradients (PCGrad) [48] alters the conflicting gradients by projecting each onto the normal plane of the other, preventing the interfering components of the gradient from being applied to the network. Specifically, it initializes the projected gradients as $\boldsymbol{v}_i^{\text{PC}} = \nabla f^i(\boldsymbol{x}_k)$. If projected gradient $\boldsymbol{v}_i$ and raw gradient $\nabla f^j(\boldsymbol{x}_k)$ are conflicting, i.e., $\boldsymbol{v}_i^\top \nabla f^j(\boldsymbol{x}_k) < 0$, then it iteratively updates the projected gradients by

$$\boldsymbol{v}_i^{\text{PC}} = \boldsymbol{v}_i^{\text{PC}}(\boldsymbol{x}_k) - \frac{\boldsymbol{v}_i^{\text{PC}}(\boldsymbol{x}_k)^\top \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|_2^2} \nabla f^j(\boldsymbol{x}_k),$$

for $i, j = 1, \ldots, m$ until the projected gradients do not conflict with the received gradients. Since the conflicting components are overall eliminated, it makes the direction $\boldsymbol{d}_k = -\frac{1}{m}\sum_{i=1}^m \boldsymbol{v}_i^{\text{PC}}$ satisfies $\boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k) < 0$ for each objective i, thus optimizing all objectives simultaneously.

Conflict-Averse Gradient (CAGrad) [30] generalizes averaging linearization and MGDA. Specifically, it constrains search region for the common direction as a circle around the average gradient $\boldsymbol{v}_0 = \sum_{i=1}^m \frac{1}{m}\nabla f^i(\boldsymbol{x}_k)$ with diagram $c\,\|\boldsymbol{v}_0\|$. Specifically, the direction $\boldsymbol{d}_k$ is yielded by

$$\boldsymbol{d}_k = \arg\min_{d\in\mathcal{R}^d} \max_{i\in[m]} \nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d}, s.t. \|\boldsymbol{d} - \boldsymbol{v}_0\| \leq c\|\boldsymbol{v}_0\|.$$

It can minimize the average loss function, while leveraging the worst local improvement of individual tasks to regularize the algorithm trajectory by using the common descent property of MGDA. Note that the gradient conflicting issue is not absolutely eliminated due to the new constraint. Still, the above objective forces direction to approach the common descent direction as close as possible. To simplify the optimization, the dual objective is

$$\boldsymbol{\lambda} = \arg\min_{\boldsymbol{\lambda}\in S_m} \left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{v}_0 + c\,\|\boldsymbol{v}_0\| \left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|.$$

Then the direction is calculated by $\boldsymbol{d}_k = -\left(\boldsymbol{v}_0 + c\,\|\boldsymbol{v}_0\| \left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right)$.

Convergence analysis. MGDA has been shown to converge to an arbitrary Pareto critical/optimal point with the same rate as single-objective optimization [11]. A similar result has been proved with PCGrad [48]. CAGrad has been shown to converge to the minimizer or stationary point of the averaging loss $\frac{1}{m}\sum_{i=1}^m f^i(\boldsymbol{x})$ when $c \in [0, 1)$, or an arbitrary Pareto critical/optimal point when $c \geq 1$ [30]. By Proposition 1, we can infer that the minimizer or stationary point of the averaging loss is also Pareto optimal or critical. Hence, we'll analyze them uniformly under the Pareto optimality in the following analysis, rather than distinguish them.

3 General Framework for Multi-Objective Gradient Manipulation

In this section, we summarize multi-objective gradient manipulation (MOGM) algorithms into a general framework with the goal of uniformly analyzing the above algorithms.

3.1 The Framework with Full-batch Gradients

Without loss of generality, we summarize the direction search of MOGM in the following framework.

$$\boldsymbol{d}_k = -\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k, \tag{2}$$

where the composite weights $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^m)^\top$ is a function of the multiple gradients $\nabla \boldsymbol{F}(\boldsymbol{x}_k) = (\nabla f^1(\boldsymbol{x}_k), \ldots, \nabla f^m(\boldsymbol{x}_k))^\top$. Specifically, MGDA and CAGrad obtain $\boldsymbol{\lambda}_k$ by their dual form of the subproblem that depends on the received gradients $\nabla \boldsymbol{F}(\boldsymbol{x}_k)$. For PCGrad, we take the two-objective case as an example. If the gradients are not conflicting, we have $\boldsymbol{\lambda}_k = (1/2, 1/2)$ as $\boldsymbol{d}_k = -\frac{1}{2}(\nabla f^1(\boldsymbol{x}_k) + \nabla f^2(\boldsymbol{x}_k))$. If they are conflicting, we can rewrite $\boldsymbol{d}_k = -\frac{1}{2}(\boldsymbol{v}_1^{\text{PC}} + \boldsymbol{v}_2^{\text{PC}})$ by extending $\boldsymbol{v}_1^{\text{PC}}$ and $\boldsymbol{v}_2^{\text{PC}}$ as the following equivalent form.

$$\boldsymbol{d}_k = -\frac{1}{2}\left(\left(1 - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{\|\nabla f^1(\boldsymbol{x}_k)\|_2^2}\right)\nabla f^1(\boldsymbol{x}_k) + \left(1 - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{\|\nabla f^2(\boldsymbol{x}_k)\|_2^2}\right)\nabla f^2(\boldsymbol{x}_k)\right).$$

Then $\boldsymbol{\lambda}_k = -(\frac{1}{2} - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{2\|\nabla f^1(\boldsymbol{x}_k)\|_2^2}, \frac{1}{2} - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{2\|\nabla f^2(\boldsymbol{x}_k)\|_2^2})$. Since PCGrad with more than three objectives should be presented as a protocol [48], we leave more details for the equivalent protocol with the general framework version in Appendix B.

Note that the composite weights $\boldsymbol{\lambda}_k$ of PCGrad and CAGrad are not constrained in the probability simplex S_m. Hence, we here consider a more general assumption on the boundness of $\boldsymbol{\lambda}_k$ for including various MOGM methods than bounding it in S_m as MGDA.

Assumption 1. *For the general framework 2, there exists a finite constant $B \in \mathbb{R}$, such that $0 \le \lambda_k^i \le B, \sum_{i=1}^m \lambda_k^i \ge 1$ for all $k = 1, \ldots, n, i = 1, \ldots, m$.*

Remark. The general framework also contains more recent approaches that are based on the fundamentals of the introduced methods, such as GradVac [44], IMTL-G [31]. Note that we focus on the optimization side for multi-objective problems, thus we provide the derivation of existing fundamental optimizers. A vertical line of works that focus on neural architecture, such as RotoGrad [18], are not included in this work.

3.2 The Framework with Stochastic Gradients

MOO can be solved with gradient manipulation algorithms with full gradients with theoretical guarantees [11, 48, 30]. However, training with mini-batch gradient is much more practical for deep learning applications [38], while MOGM with stochastic gradients is largely underexplored. Specifically, we only obtain noisy gradient feedback $\boldsymbol{G}_k = \left(\boldsymbol{g}^1(\boldsymbol{x}_k, \boldsymbol{\xi}_k^1), \ldots, \boldsymbol{g}^m(\boldsymbol{x}_k, \boldsymbol{\xi}_k^m)\right)$ that is corrupted by stochastic noise $\boldsymbol{\xi}_k^i, i = 1, \ldots, m$. Following 2, we have its stochastic version

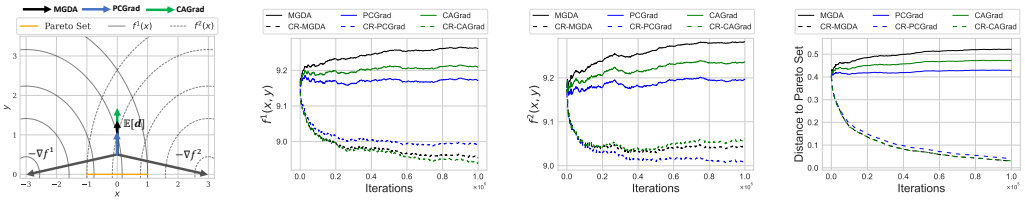
$$\boldsymbol{d}_k = -\boldsymbol{G}_k\boldsymbol{\lambda}_k. \tag{3}$$

Note that here $\boldsymbol{\lambda}_k$ depends on stochastic gradients, not the exact ones. We then make the following standard assumption similar to single objective setting [20, 14], which says that we have access to unbiased stochastic estimates of the gradients for all objectives.

Assumption 2. *Let $\{\boldsymbol{x}_k\}_{k=1}^n$ be the iterate sequence for any algorithm. For all $k = 1, \ldots, n$, we have access to $\boldsymbol{g}^1(\boldsymbol{x}_k, \boldsymbol{\xi}_k^1), \ldots, \boldsymbol{g}^m(\boldsymbol{x}_k, \boldsymbol{\xi}_k^m)$ which are unbiased estimates of $\nabla f^1(\boldsymbol{x}_k), \ldots, \nabla f^m(\boldsymbol{x}_k)$, i.e. $\mathbb{E}_{\boldsymbol{\xi}_k}\left[\boldsymbol{g}^i(\boldsymbol{x}_k, \boldsymbol{\xi}_k^i)\right] = \nabla f^i(\boldsymbol{x}_k), i = 1, \ldots, m$. Further, each gradient variance is bounded by $\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{g}^i(\boldsymbol{x}_k, \boldsymbol{\xi}_k^i) - \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \le \sigma^2, i = 1, \ldots, m$.*

To analyze MOGM in the stochastic setting, we state the following definition to determine a sequence generated by an iterative algorithm to asymptotic converge to Pareto critical and optimal.

Definition 3 (**Convergence condition**). ***(a)*** *A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotically converges to Pareto critical points if $\lim_{k\to\infty} \mathbb{E}[\min_{\boldsymbol{\lambda}_k^* \in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|] \to 0$.*
(b) *For convex multi-objective function $\boldsymbol{F}(\cdot)$. A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotically converges to Pareto optimal if $\lim_{k\to\infty} \mathbb{E}[\max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} (\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))] \to 0$.*



(a) Inverse Direction (b) Loss 1 (c) Loss 2 (d) Distance to Optimal

Figure 1: (a) We plot the contours of two objectives. Gradient manipulation algorithms designed based on full-batch gradient could yield an expected composite gradient that conflicts with the true gradients and is inverse to Pareto set. (b) (c) This leads to the unexpected consequence of degrading all the objectives, making the objectives insufficiently optimized. (d) Hence, MGDA, PCGrad, and CAGrad can not converge to Pareto optimal solution, and suffer a large distance to Pareto optimal set.

The definition (a) directly follows Proposition 1 (a) and the definition of Pareto critical. The definition (b) is because $\boldsymbol{x}_k^*$ is the minimum of ${\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x})$, thus is Pareto optimal (Proposition 1 (b)). Asymptotically converging to ${\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*)$ implies that $\boldsymbol{x}_k$ approaches Pareto optimal. Note that the minimization operator of $\boldsymbol{\lambda}_k^*$ is from the existence condition in Proposition 1 (b). This definition can also be viewed as the stochastic version of metric function [42] and Pareto suboptimal gap [19].

4 The Non-convergence of Stochastic Multi-Objective Gradient Manipulation

With the problem setup of stochastic MOGM in the previous section, we now discuss fundamental flaws among the current gradient manipulation methods that are all designed and analyzed based on full-batch gradients. We show that such batch algorithms fail to converge to a Pareto optimal solution even in a simple two-objective two-dimensional convex stochastic setting. The main issue lies in the strong dependence of $\boldsymbol{\lambda}_k$ on the received stochastic gradients $\boldsymbol{G}_k$. Surprisingly, we find that combining such correlated $\boldsymbol{\lambda}_k$ with the stochastic gradient yields a biased composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k\boldsymbol{\lambda}_k]$ that even conflicts with all full-batch gradients in expectation. This inverse direction even causes the algorithms to inversely optimize all the objectives simultaneously.

An illustrative example. Next, we provide a simple problem instance in which MGDA, PCGrad, CAGrad can not converge to Pareto optimal. Consider the following two-dimensional stochastic optimization setting over the domain $\mathcal{K} = \{(x, y) \mid x \in [-1, 1], y \geq 0\}$, where p represents the probability of occurrence

$$\tilde{f}^1(x, y) = \begin{cases} (x+1)^2 + (y-2)^2 & p = 1/2 \\ (x+5)^2 + (y+2)^2 & p = 1/2 \end{cases}, \tilde{f}^2(x, y) = \begin{cases} (x-1)^2 + (y-2)^2 & p = 1/2 \\ (x-5)^2 + (y+2)^2 & p = 1/2 \end{cases}. \quad (4)$$

The expected function is $f^1(x, y) = (x+3)^2 + y^2 + 8$, $f^2(x, y) = (x-3)^2 + y^2 + 8$. Thus, by ignoring the constant term, the optimization goal is to minimize the distance towards $(-3, 0)$ and $(3, 0)$ simultaneously. We can easily know that the Pareto set is the line segment $\{(a, 0) \mid a \in [-1, 1]\}$.

In the first step, by calculating the expected gradient of the point $(0, 0.4)$, we find that all the gradient manipulation algorithms generate a vertical-up direction in expectation, as shown in Figure 1a, which moves the algorithm in the wrong direction away from Pareto optimal setting. More seriously, the expected direction $\mathbb{E}[\boldsymbol{d}_k] = (0, \epsilon), \epsilon > 0$ is conflicting with both true gradients, suggesting that it inversely optimizes both objectives.

Furthermore, we could prove that the inverse direction problem happens with Pareto optimal points, which resists the algorithms to approach Pareto optimal. Using the intuition above, we show that gradient manipulation algorithms can not converge to Pareto optimal for this setting.

Theorem 1. *There is a stochastic convex optimization problem for which MGDA, PCGrad, CAGrad do not converge to the Pareto optimal solution.*

We relegate the proof to Appendix D. In the proof, we provide stronger support that at least in the range $\{(x, y) \mid x \in [-1, 1], y \in [0, 0.05]\}$, the expected directions for the three typical algorithms all move the solution away from Pareto set. This means there exists a constant gap for gradient

manipulation algorithms to approach the Pareto optimal set. Note that this gap is a sufficient condition to verify the non-convergence, and it can be much larger, as shown in Figure 1d.

Intuitively, the convergence gap is caused by the strong correlation between the composite weights $\boldsymbol{\lambda}_k$ and the stochastic gradients $\boldsymbol{G}_k$, i.e., $\boldsymbol{\lambda}_k$ is solely determined by $\boldsymbol{G}_k$, which could produce an inverse direction in the example above. When the composite weights is independent of the stochastic gradients, the expected composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k\boldsymbol{\lambda}_k] = -\nabla F(\boldsymbol{x}_k)\boldsymbol{\lambda}_k$ will be a positive direction to approach the Pareto set. Hence, methods like linear scalarization are able to attain a point in the Pareto optimal set [13], but are deprived of the practical benefits of gradient manipulation.

We would also like to emphasize that while the example of non-convergence is carefully constructed to demonstrate the problems in the above three typical algorithms more easily, such an issue also happens with more general Gaussian noise, as shown in Appendix G.2. Hence, it is not unrealistic to suspect scenarios where such an issue can impair practical performance. It is also reasonable to assume that this problem is especially aggravated in high dimensional settings, where the variance of the gradients is more significant.

Comparison with existing convergence results. As introduced in Section 2.2, the optimal solution for the multi-task objective $\frac{1}{m}\sum_{i=1}^{m} f^i(\boldsymbol{x})$ must be Pareto optimal. Therefore, the above non-convergence analysis generates well with MOGM algorithms that optimize the averaging loss, such as CAGrad. This result shows that the previous convergence analyses in the full-batch setting are all invalid in the stochastic setting.

To the best of our knowledge, there is only MGDA that has convergence analysis in the stochastic setting. Liu et al. [33] establishes the convergence of MGDA with stochastic gradients, but based on two unjustified assumptions. The first is that the gradient variance asymptotically converges to 0. The second is that the subproblem for $\boldsymbol{\lambda}_k$, denoted as function $\boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g}^m)$, is Lipschitz continuous with right to each gradient. The first assumption needs to gradually increase the batch size to full batch; however, the reliance on giant batch sizes is impractical for deep learning training [4]. The second one can significantly simplify the convergence analysis, but is actually not true as shown in the following proposition.

Proposition 2. *Subproblem $\boldsymbol{\lambda}_k(\cdot)$ is not Lipschitz continuous with right to each gradient for MGDA, i.e., there exist no finite β such that $\|\boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g}^i, \ldots, \boldsymbol{g}^m) - \boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g'}^i, \ldots, \boldsymbol{g}^m)\| \leq \beta\|\boldsymbol{g}^i - \boldsymbol{g'}^i\|$ for $i = 1, \ldots, m$ and gradient $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^i, \boldsymbol{g'}^i, \ldots, \boldsymbol{g}^m$.*

Similar results can be proved for PCGrad and CAGrad, and we relegate the proof to the appendix D.

5 Correlation-Reduced Stochastic Multi-Objective Gradient Manipulation

In this section, we develop a new principled exponential moving average mechanism on the composite weights $\boldsymbol{\lambda}_k$, and provide its theoretical benefit for fixing the convergence issue for MOGM algorithms. Our aim is to devise a new strategy with guaranteed convergence, while preserving the practical benefits of MOGM without introducing additional computation and memory overhead.

Algorithmic insight. To understand the design of the mechanism, recall that the main issue of the non-convergence for MOGM lies in the strong correlation between the composite weights $\boldsymbol{\lambda}_k$ and the stochastic gradients $\boldsymbol{G}_k$, which could cause the composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k\boldsymbol{\lambda}_k]$ inversely optimizing all the objectives in expectation. It is natural to consider reducing such correlation, which can be bounded by the following lemma.

Lemma 1. *Under Assumption 1, 2. The correlation between $\boldsymbol{\lambda}_k$ and $\boldsymbol{G}_k$ can be bounded as $\|\mathbb{E}[\boldsymbol{G}_k\boldsymbol{\lambda}_k] - \mathbb{E}[\boldsymbol{G}_k]\mathbb{E}[\boldsymbol{\lambda}_k]\|_2^2 \leq mB^2\sigma^2\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k)$, where $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) = \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{\lambda}_k - \mathbb{E}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k]\|_2^2]$.*

From Lemma 1, we know that the correlation can be reduced by reducing the variances of $\boldsymbol{\lambda}_k$. From this intuition, we can modify MOGM algorithms to reduce such correlation by regularizing the variation of $\boldsymbol{\lambda}_k$. Correlation-reduced MOGM (CR-MOGM) adopts a momentum mechanism with coefficient α_k on the update of composite weights as Line 5 in Algorithm 1, which is shown to be able to reduce variance as

Lemma 2. *Under Assumption 1, $\boldsymbol{\lambda}_k$ variance for Algorithm 1 is bounded $\mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k] \leq m^2B^2(1-\alpha_k)^2$.*

This result shows that the correlation can be controlled by choosing α_k closed enough to 1. With a reduced correlation, the wrong direction generated by MOGM could be corrected to a positive

Algorithm 1 Correlation-Reduced Stochastic Multi-objective Gradient Manipulation (CR-MOGM)

Input: Number of iterations K, regularization parameter α_k, learning rate η_k.
Initialize: x_0, $\boldsymbol{\lambda}_0 \in \Delta_m$.
for $k = 1, \ldots, K$ **do**
  Compute $\hat{\boldsymbol{\lambda}}_k$ by gradient manipulation algorithms using the stochastic gradients.
  Update the weights for the gradient composition $\boldsymbol{\lambda}_k = \alpha_k \boldsymbol{\lambda}_{k-1} + (1-\alpha_k)\hat{\boldsymbol{\lambda}}_k$.
  Compute the composite gradient $\boldsymbol{d}_k = -\boldsymbol{G}_k(\boldsymbol{x}_k, \xi)\boldsymbol{\lambda}_k$.
  Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \boldsymbol{d}_k$.
end for

direction, and thus the algorithm would be able to converge. Note that the momentum on $\boldsymbol{\lambda}_k$ only requires an additional negligible computation cost.

On the other hand, the proposed exponential moving average mechanism also preserves the advantage of MOGM by carefully setting the momentum parameter α_k to be 0 and asymptotically convergent to 1. Intuitively, CR-MOGM performs similarly to vanilla MOGM at the initial stage, and approximates like fixed linear scalarization in the late stage. Initially, the stochastic noise occupies a relatively small component of the stochastic gradient. Thus, methods like MGDA, PCGrad, CAGrad can search for good common descent directions. When the number of iterations increases and the solution approaches the Pareto optimal, the impact of the stochastic noise stands out, causing the inverse direction problem for MOGA as in the previous section. At this time, α_k becomes larger, maintaining relatively stable $\boldsymbol{\lambda}_k$ eliminates the strong correlation and guarantees the convergence.

Convergence analysis. We next provide convergence analysis for CR-MOGM 1 concluding all algorithms based on the general framework 3. Before that, we introduce basic assumptions.

Assumption 3. *$f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x})$ are all differentialable, H-Lipschitz and L-smoothness, suggesting that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}, i \in [m]$, it holds $\|\nabla f^i(\boldsymbol{y})\| \leq H$ and $\|\nabla f^i(\boldsymbol{x}) - \nabla f^i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.*

Lipschitz and smoothness are commonly used in complexity analysis for stochastic gradient algorithms [20, 14, 12, 4]. Based on Definition 3, we prove the following key results for CR-MOGM.

Theorem 2 (Convex). *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x})$ are all convex with bounded optimal values as $f^1(\boldsymbol{x}^*), \ldots, f^m(\boldsymbol{x}^*) \leq F$ for $\boldsymbol{x}^*$ in Pareto set, and the distance from sequence to Pareto set is bounded, i.e., $\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq D$. Set $0 \leq \eta_n \leq \ldots \leq \eta_1 \leq 1/mLB, \alpha_k \in (0,1]$ for $k = 1, \ldots, n$ in Algorithm 1, it achieves*

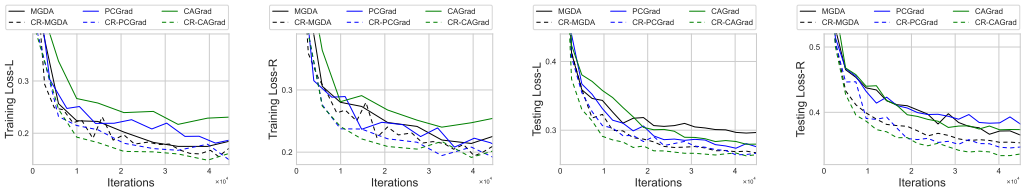
$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} (\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right] \leq \frac{D^2}{\eta_n n} + \frac{m^2B^2(\sigma^2 + \sigma H + H^2)}{n}\sum_{k=1}^{n}\eta_k$$
$$+ \frac{DB\sigma m^{3/2}}{n}\sum_{k=1}^{n}(1-\alpha_k) + \frac{6m^{5/2}B^2H\sigma}{n}\sum_{k=1}^{n}(1-\alpha_k)\eta_k + \frac{2F}{n}\sum_{l=1}^{n} l(1-\alpha_{l+1})\sum_{i=1}^{m}|\lambda_l^i - \hat{\lambda}_{l+1}^i|.$$

Theorem 3 (Non-convex). *Under Assumption 1, 2, 3. Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}_k), \ldots, f^m(\boldsymbol{x}_k) \leq F$ are all bounded for $k = 1, \ldots, n$. Set $0 \leq \eta_n \leq \ldots \leq \eta_1 \leq 1/mLB$ and $\alpha_k \in (0,1]$ in Algorithm 1, it achieves*

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^* \in S_m}\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right] \leq \frac{F}{n}\sum_{k=2}^{n}\frac{2}{\eta_k}(1-\alpha_k)\sum_{i=1}^{m}|\hat{\lambda}_k^i - \lambda_{k-1}^i| + \frac{2mBF}{n\eta_n}$$
$$+ \frac{4m^{5/2}B^2H\sigma}{n}\sum_{k=1}^{n}(1-\alpha_k) + \frac{Lm^3B^3\sigma^2}{2n}\sum_{k=1}^{n}\eta_k.$$

Remark. Note that our theoretical results imply that CR-MOGM is convergent in averaging by choosing suitable η and α. Such an averaging analysis scheme is adopted by CAGrad [30], and is often used in stochastic optimization [41, 15, 4], e.g. Adagrad [28], Adam [24, 38]. It can be transformed into the traditional version, as shown in Appendix E.7

The above results falls as an immediate corollary of the above Theorem.



(a) Training Loss-L (b) Training Loss-R (c) Testing Loss-L (d) Testing Loss-R

Figure 2: Performance comparison of gradient manipulation algorithms and their correlation-reduced counter part for MultiMNIST. We report the mean over 3 runs.

Table 1: Accuracy on MultiMNIST. We report the means over 3 runs.

Method	task-L Acc	task-R Acc
MGDA	96.46%	95.29%
CR-MGDA	**96.92**% ↑	**95.44**% ↑
PCGrad	96.63%	95.37%
CR-PCGrad	**96.90**% ↑	**95.75**% ↑
CAGrad	96.47%	95.20%
CR-CAGrad	**96.85**% ↑	**95.70**% ↑

Table 2: Losses of the final output of MGDA and CR-MGDA on MultiMNIST. We report the mean over 3 runs.

	MGDA		CR-MGDA	
Batch Size	loss-L	loss-R	loss-L	loss-R
8	0.44	0.54	**0.24**	**0.35**
4	0.57	0.65	**0.26**	**0.36**
2	0.71	0.81	**0.28**	**0.40**

Corollary 1. *Set* $1 \geq \eta_k = O(1/\sqrt{k}), \alpha_k = \max\{1 - \eta_k/\eta_1, 1 - \eta_k/\eta_1(k\sum_{i=1}^m |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ *in Theorem 2,3, Algorithm 1 converges with* $O(n^{1/2})$, $O(n^{1/4})$ *rates in the convex and non-convex case.*

This result indicates that Algorithm 1 has a convergence rate that matches the one in the single-objective stochastic setting [7, 2]. It demonstrates the efficacy of the proposed momentum mechanism.

6 Experiments

In this section, we present empirical results on both synthetic with convex objectives and real-world datasets for multi-task learning with deep neural networks, representing convex and nonconvex settings, respectively. We apply CR-MOGM to MGDA, PCGrad, CAGrad, and use CR- to distinguish whether it adopts our approach. We conduct experiments to answer the following questions:

Q1: Does our approach fix the non-convergence and improve the optimality for MOGM algorithms?

Q2: Does our approach consistently improve the practical performance of MOGM in deep learning?

Q3: Does the non-convergence issue demonstrated with the special case still exists in practice?

Q4: Is our approach flexible to be implemented and agnostic to models as well as optimizers?

Synthetic experiments. To demonstrate whether CR-MOGM fixes the convergence issue for gradient manipulation algorithm, we first consider the same simple convex setting as the example 4 in Section 4. We initialize $\boldsymbol{x}_0 = (0, 0.4)$ and use SGD as the optimizer. To enable fair comparison, the adaptive stepsize is set with $\eta_k = 0.006/\sqrt{k}$ for all the algorithms, and the momentum parameter for smoothing the composite weights is set with the suggested parameter setting in Corollary 1.

Figure 1 illustrates the multi-objective losses and distance to the Pareto set for this problem. We first note that both the two objective losses of MGDA, PCGrad, CAGrad are significantly larger than their counterparts applied with the proposed mechanism. Furthermore, the losses are even increasing at the initial stage as a result of the inverse direction issue demonstrated in Figure 1a. It moves $\boldsymbol{x}_k$ away from the Pareto set, which unfortunately causes an increasing suboptimal gap. In contrast, applying CR-MOGM enhances the algorithms to converge to Pareto optimal. This positively answers **Q1**.

Multi-task classification. For multi-task classification scenarios, we follow the experiment setup from Sener et al. [40], and consider the MultiMNIST dataset [39] with 60K examples. MultiMNIST contains two tasks: classifying the digit on the top-left (task-L) and classifying the digit on the bottom-

Method	Segmentation (Higher Better)				Depth (Lower Better)			
	(Training)		(Testing)		(Training)		(Testing)	
	mIoU	Pix Acc	mIoU	Pix Acc	Abs Err	Rel Err	Abs Err	Rel Err
MGDA [40]	76.05	94.08	66.32	88.85	**0.0065**	18.90	0.0248	**33.50**
CR-MGDA	**79.58**	**95.12**	**73.21**	**93.01**	0.0067	**18.51**	**0.0151**	33.59
PCGrad [48]	**80.19**	95.31	75.25	**93.50**	0.0087	27.18	**0.0148**	45.37
CR-PCGrad	80.12	**95.32**	**75.37**	**93.50**	**0.0085**	**26.44**	0.0153	**43.05**
CAGrad [30]	80.24	95.29	75.09	93.52	0.0081	**25.06**	0.0145	40.97
CR-CAGrad	**80.33**	**95.33**	**75.58**	**93.53**	**0.0080**	25.30	**0.0142**	**38.51**

Table 3: Multi-task learning results on CityScapes Challenge. We report the mean over 3 runs.

right (task-R). We use the LeNet architecture [25], and use SGD as the optimizer. The stepsize and parameters are tuned from vanilla MGDA, PCGrad, CAGrad, and the correlation-reduced counterparts follow the same parameter setting to enable a fair comparison. The momentum parameter is set with suggested value in Corollary 1.

We report the train and test losses in Figure 2. We can see that algorithms with $\boldsymbol{\lambda}$ momentum perform better than original ones with respect to both train and test losses. We also observed that $\boldsymbol{\lambda}$ momentum consistently improves the accuracy for all the objectives in Table 1. We hence positively answer **Q2**.

In particular in Table 2, we observe that the loss of vanilla MGDA becomes significantly larger when the batch size is smaller, while CR-MGDA appears not sensitive to the batch size. The possible reason is that the convergence issue of MOGM becomes more severe with larger stochastic noise, which severely degrades the performance of MOGM. Since CR-MOGM is proven to converge to the Pareto optimal/critical, it is much more robust to stochastic noises. Thus, in this regard, our method is more practical for real-world applications. We hence positively answer **Q3**.

Semantic segmentation. For semantic segmentation scenarios, we follow the experiment setup from Yu et al. [48] and Liu et al. [30], and consider the CityScapes. CityScapes similarly contains two tasks: 7-class semantic segmentation and depth estimation. We combine algorithms with MTAN [32] based on SegNet architecture [1]. We use adam [24] as the optimizer. All parameters are tuned from vanilla MOGM algorithms, and the correlation-reduced counterparts follow the same parameter setting to enable a fair comparison. The momentum parameter is set with the suggested value.

We report mean Intersection-Over-Union (mIoU), pixel accuracy (Pix Acc), and absolute/relative (abs/rel) errors in Table 3. We observe comparable or better performance when adopting our approach with the three gradient manipulation methods. In particular, the improvements for MGDA and CAGrad are more significant than PCGrad, possibly due to the special gradient manipulation rule that PCGrad does not alter the gradient when they are not conflicting, then the non-convergence issue of PCGrad could be less severe. Combining the above results, we demonstrate that our methods work for different models and optimizers. We hence give a positive answer to **Q4**.

7 Conclusions

In this work, we study multi-objective gradient manipulation in the stochastic mini-batch setting and identify a fundamental flaw in algorithms designed under the exact full-batch setting. Through an illustrative example, we rigorously prove the non-convergence issue of typical methods, including MGDA, PCGrad, and CAGrad. In particular, we fix this problem by slightly changing these algorithms by an exponential smoothing mechanism without introducing additional computational or memory overhead. We prove the convergence of the resulting algorithms and show that they yield consistent empirical performance improvement. Our analysis is general and can be extended to analyze combining MOGM algorithms with other stochastic optimizers such as Adam and Adagrad.

Limitations. This paper discusses the typical MOGM algorithms with strict convergence guarantees, i.e., MGDA, PCGrad, and CAGrad. Other empirical-driven methods without rigorous theoretical guarantees, such as RotoGrad [18], are too expensive to be included in our analysis. We believe that the convergence issue in that line of literature will be an excellent future work to explore in depth.

Acknowledgement

This work is supported by the National Key Research and Development Program of China No. 2020AAA0106300 and National Natural Science Foundation of China No. 62250008. This work was supported by Ant Group through Ant Research Intern Program. We would like to thank Tianyu Liu, Xin Wang, and Ziwei Zhang for their advices on paper writing.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[3] Ana Luísa Custódio, JF Aguilar Madeira, A Ismael F Vaz, and Luís Nunes Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

[4] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Kalyanmoy Deb. Multi-objective optimization. In *Search Methodologies*, pages 403–449. Springer, 2014.

[6] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[7] Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In *International Conference on Machine Learning*, pages 2658–2667, 2020.

[8] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

[9] Benjamin Fehrman, Benjamin Gess, and Arnulf Jentzen. Convergence rates for the stochastic gradient descent method for non-convex objective functions. *Journal of Machine Learning Research*, 21:136, 2020.

[10] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[11] Jörg Fliege, A Ismael F Vaz, and Luís Nunes Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019.

[12] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209, 2019.

[13] Walter J Gutjahr and Alois Pichler. Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, 236(2):475–499, 2016.

[14] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

[15] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[16] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, 2019.

[17] Johannes Jahn. *Vector Optimization: Theory, Applications, and Extensions*. Springer Science & Business Media, 2010.

[18] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022.

[19] Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Lihong Gu, Xiaodong Zeng, and Wenwu Zhu. Multi-objective online learning. 2021.

[20] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

[21] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[22] Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

[23] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34:30271–30283, 2021.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[26] Yunwen Lei, Ting Hu, Guiying Li, and Ke Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[27] Kfir Levy, Ali Kavis, and Volkan Cevher. Storm+: Fully adaptive sgd with recursive momentum for nonconvex optimization. *Advances in Neural Information Processing Systems*, 34:20571–20582, 2021.

[28] Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *International Conference on Artificial Intelligence and Statistics*, pages 983–992, 2019.

[29] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[30] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[31] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[32] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[33] Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pages 1–30, 2021.

[34] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[35] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018.

[36] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *Conference on Research & Development in Information Retrieval*, pages 1137–1140, 2018.

[37] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[38] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[39] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in Neural Information Processing Systems*, 30, 2017.

[40] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 525–536, 2018.

[41] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

[42] Hiroki Tanabe, Ellen H Fukuda, and Nobuo Yamashita. Proximal gradient methods for multi-objective optimization and their applications. *Computational Optimization and Applications*, 72(2):339–361, 2019.

[43] Hiroki Tanabe, Ellen H Fukuda, and Nobuo Yamashita. Convergence rates analysis of a multiobjective proximal gradient method. *Optimization Letters*, pages 1–18, 2022.

[44] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2020.

[45] Ruobing Xie, Yanlei Liu, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. Personalized approximate pareto-efficient recommendation. In *Web Conference*, pages 3839–3849, 2021.

[46] Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *International Joint Conference on Artificial Intelligence*, pages 2955–2961, 2018.

[47] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

[48] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[49] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019.

[50] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Supplementary Materials for "On the Convergence of Stochastic Multi-Objective Gradient Alteration and Beyond"

We organize the supplemental materials as follows. Section A provides a detailed comparison between weak Pareto optimal and Pareto optimal. Section B provides the derivation of PCGrad with more than two objectives to our general framework. Section C discusses more details of the convergence analysis framework in Definition 3. Section D supplements the theoretical results on the non-convergence for MOGA algorithms. Section E supplements the convergence results for both convex (Theorem 2) and non-convex (Theorem 3) cases. In particular, we discuss the averaging scheme of the convergence theorem in more detail, and provide a way to transform it to traditional convergence using a typical uniformly sampling technique in stochastic convergence analysis [4, 26, 7, 22]. Section F provides the convergence analysis for the cases with strongly convex functions. Section G introduces additional details of the experiments, and provides an additional simulation result with Gaussian noises.

A Detailed Comparison between Weak Pareto Optimal and Pareto Optimal

In this section, we give a rigorous comparison between the definitions of Pareto optimal and weak Pareto optimal, where the latter is defined as

Definition 4 (Weak Pareto optimal)**.** *A solution $\boldsymbol{x}^* \in \mathcal{K}$ is called weak Pareto optimal if there is not $x \in \mathcal{K}$ such that $F(\boldsymbol{x}) < F(\boldsymbol{x}^*)$, i.e. $f^i(\boldsymbol{x}) \leq f^i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, m$.*

By the Definition 1 and Definition 4, it can be easy to know that Pareto optimal points are always weak Pareto optimal, and the converse is not always true. Take a two-objectives example, the minimal points for f^1 are all Pareto weak optimal, but only ones that are also optimal for f^2 are Pareto optimal. Furthermore, if we assume f^1 has a unique minimizer, which is then Pareto optimal.

In rigorous words, we state below that relation among Pareto optimal and weak Pareto optimal, as a rigorous version for Proposition 1 (b).

Proposition 3 (Theorem 5.13 and Lemma 5,14 in [17])**.** *For MOO problem 1, the following statements hold:* ***(a)*** *For convex objectives $\boldsymbol{F}(\cdot)$. If there exists $\boldsymbol{\lambda} \in S_m$ such that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{K}} \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x})$, then $\boldsymbol{x}^*$ is weak Pareto optimal.* ***(b)*** *Further if $\boldsymbol{x}^*$ is the unique minimizer, then $\boldsymbol{x}^*$ Pareto optimal.*

For ease to understand, previous studies [11, 33, 30] do not distinguish the detailed difference between the above definitions. This paper also follows this presentation trick.

B More Details of the General Framework for MOGA

PCGrad with more than 2 objectives. Specifically, the PCGrad is initialized by $\boldsymbol{v}_i^{\text{PC}} = \nabla f^i(\boldsymbol{x}_k)$. When subtracting the conflicting component, i.e., $\boldsymbol{v}_i^{\text{PC}} = \boldsymbol{v}_i^{\text{PC}} - \frac{\boldsymbol{v}_i^{\text{PC}\top} \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|^2} \nabla f^j(\boldsymbol{x}_k)$, it is equivalent to subtracting a weight $\frac{\boldsymbol{v}_i^{\text{PC}\top} \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|^2} \nabla f^j(\boldsymbol{x}_k)$ for gradient $\boldsymbol{v}_j$ if using the framework 2. Provided with such equivalence, $\boldsymbol{d}_k = -\frac{1}{m} \sum_{i=1}^m \boldsymbol{v}_i^{\text{PC}}$ is equivalent to $\boldsymbol{d}_k = -\nabla \boldsymbol{F}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k$ using Algorithm 2

Algorithm 2 Calculation for composite weight $\boldsymbol{\lambda}_k$ in PCGrad

Input: $\nabla f^i(\boldsymbol{x}_k), i = 1, \ldots, m$
Initialize: $v_i = \nabla f^i(\boldsymbol{x}_k), v_i^{\text{PC}} = v_i, \lambda^i = 1$ for $i = 1, \ldots, m$
for $i \in [m]$ **do**
  for $j \overset{\text{uniformly}}{\sim} [m] \setminus i$ in random order **do**
    if ${v_i^{\text{PC}}}^\top \nabla f^j(\boldsymbol{x}_k) < 0$ **then**
      Set $v_i^{\text{PC}} = v_i^{\text{PC}} - \frac{{v_i^{\text{PC}}}^\top \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|^2} \nabla f^j(\boldsymbol{x}_k)$
      Update $\lambda^j = \lambda^j - \frac{{v_i^{\text{PC}}}^\top \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|^2}$
    end if
  end for
end for
return $\boldsymbol{\lambda}$

C More Details of the Convergence Analysis Framework

In this section, we provide more details about our stochastic convergence analysis framework. We first offer proof details of the Proportion 1, and then generalize previous analysis based on it. Finally, we give a discussion for Definition 3 that is the stochastic convergence framework based on the previous full-batch setting.

C.1 Proof of Proportion 1

Proof. The Pareto criticality can be equivalently defined as: a $\boldsymbol{x}^* \in \mathcal{K}$ is Pareto critical, if $\min_{\boldsymbol{\lambda}^* \in S_m} \|\sum_{i=1}^m {\lambda^*}^i \nabla f^i(\boldsymbol{x}^*)\| = 0$, where $S_m = \{(\lambda^1, \ldots, \lambda^m) | \sum_{i=1}^m \lambda^i = 1, \lambda \in [0, 1]\}$ denotes the probability simplex. Therefore, if there exist a $\boldsymbol{\lambda} \in S_m$, such that $0 \leq \|\sum_{i=1}^m \lambda^i \nabla f^i(\boldsymbol{x}^*)\| = 0$, then $\min_{\boldsymbol{\lambda}^* \in S_m} \|\sum_{i=1}^m {\lambda^*}^i \nabla f^i(\boldsymbol{x}^*)\| \leq \|\sum_{i=1}^m \lambda^i \nabla f^i(\boldsymbol{x}^*) = 0$. We thus prove Proposition **(a)**, and Proposition **(b)** directly comes from Proposition 3. □

C.2 More Details of Previous Convergence Analysis in Full-batch Gradient Setting

Previous convergence analyses in full-batch gradient setting, including MGDA [11, 42], PCGrad [48], CAGrad [30] can be analyzed based on Proposition 1 that reflects Pareto optimality and criticality, which induces the following convergence analysis framework.

Definition 5. ***(a)*** *A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotically converges to Pareto critical points if* $\lim_{k \to \infty} \min_{\boldsymbol{\lambda}_k^* \in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k^*\| \to 0$.
(b) *For convex multi-objective function $\boldsymbol{F}(\cdot)$. A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotically converges to Pareto optimal if* $\lim_{k \to \infty} \max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} ({\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*)) \to 0$.

(a) can be reviewed as the limit for attaining the criticality condition (the equivalent definition in [30]). **(b)** is equivalent to that there exists a composite weight $\boldsymbol{\lambda}_k^* \in S_m$ such that ${\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - \min_{\boldsymbol{x}_k^* \in \mathcal{K}} {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*) \to 0$, which is essentially the limit for attaining Pareto optimal condition in Proposition 1. It can be also reviewed the the convergence for multiobjective metric function [42].

We next briefly introduce the analysis of MGDA [11, 42] as an typical example that provides both nonconvex and convex convergence. It proves that the norm of min-norm direction $\min_{\boldsymbol{\lambda}_k^* \in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k^*\|$ is asymptotically convergent to zero with an order of $O(1/\sqrt{n})$, which is reviewed as the convergence rate with non-convex functions.

Theorem 4 (Theorem 3.1 in [11])**.** *For MGDA algorithm updating with stepsize $\eta \leq 1/L$*

$$\frac{1}{n} \sum_{k=1}^n \min_{\boldsymbol{\lambda}_k^* \in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k^*\| \leq O(1/\sqrt{n})$$

This result also implies that the composite gradient $\boldsymbol{d}_k = \|\nabla \boldsymbol{F}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k^*\|$ for MGDA asymptotically vanishes, which indicates that the sequence $\boldsymbol{x}_k$ iterated by gradient descent using $\boldsymbol{d}_k$ limits to a critical point $\boldsymbol{x}^*$.

If the multi-objective function is convex for each objective, such a critical point $\boldsymbol{x}^*$ is also weak Pareto optimal [43]. It is also not difficult to verify that if $\boldsymbol{x}^*$ is a Pareto stationary point, then it satisfies $\nabla \boldsymbol{F}(\boldsymbol{x}^*)\boldsymbol{\lambda}^* = 0$ for one $\boldsymbol{\lambda}^* \in S_m$. If the functions are convex, this meets the optimal condition of the composite function ${\boldsymbol{\lambda}^*}^\top \boldsymbol{F}(\boldsymbol{x}^*)$. And by Proposition 1, we then know that $\boldsymbol{x}$ is weak Pareto optimal.

It can prove that $\sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - \sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}^*) \leq O(1)$. Since in the batch-gradient setting, $\boldsymbol{d}_k$ is the common descent direction such that $\boldsymbol{F}(\boldsymbol{x}_k)$ is decreasing with respect to k. Thus it has $\sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_n) - \sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}^*) \leq \sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - \sum_{k=1}^n {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}^*) \leq O(1)$. Therefore, we have the following theorem

Theorem 5 (Theorem 4.1 in [11])**.** *For MGDA algorithm updating with stepsize $\eta \leq 1/L$*

$$\left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x}_n) - \left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x}^*) \leq O(1/k).$$

Where the averaging weight $\frac{\lambda_k^*}{n}$ is also in S_m. Since in the proof, the comparing point $\boldsymbol{x}^*$ can be generated to arbitrarily chosen $\boldsymbol{u} \in \mathcal{K}$ [43]. We can further generate the result with $\left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x}_n) - \left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{u}) \leq O(1/k)$. By letting $\boldsymbol{u} = \arg\min_{\boldsymbol{x}\in\mathcal{K}} \left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x})$, it finally has

$$\begin{aligned}
&\max_{\boldsymbol{x}_k^*\in\mathcal{K}} \min_{\boldsymbol{\lambda}_k^*\in S_m} ({\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*)) \\
&\leq \left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x}_n) - \left(\frac{\sum_{k=1}^n \boldsymbol{\lambda}_k^*}{n}\right)^\top \boldsymbol{F}(\boldsymbol{x}^*) \leq O(1/k).
\end{aligned}$$

Similarly, PCGrad [48] is proved to converge at a Pareto critical point (Theorem 1 in [48]). CAGrad [30] generalizes the analysis of MGDA, and shows that it converges to the Pareto critical point that tends to be the critical point of the average loss when $c \in [0, 1)$. These convergence results can all be analyzed under the Pareto convergence framework as Definition 5.

C.3 More Discussion for Definition 3

Definition 3. ***(a)*** *A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotic converges to Pareto critical points if $\lim_{k\to\infty} \mathbb{E}[\min_{\boldsymbol{\lambda}_k^*\in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|] \to 0$.*
(b) *For convex multi-objective function $\boldsymbol{F}(\cdot)$. A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotic converges to Pareto optimal if $\lim_{k\to\infty} \mathbb{E}[\max_{\boldsymbol{x}_k^*\in\mathcal{K}} \min_{\boldsymbol{\lambda}_k^*\in S_m} ({\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*))] \to 0$.*

Definition 3 is a stochastic version for Definition 5. Similarly, **(a)** can be reviewed as the expected limit for attaining the criticality condition. **(b)** is equivalent to that there exists a composite weight $\boldsymbol{\lambda}_k^* \in S_m$ such that $\mathbb{E}[{\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k) - \min_{\boldsymbol{x}_k^*\in\mathcal{K}} {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*)] \to 0$, which means ${\boldsymbol{\lambda}_k^*}^\top \mathbb{E}[\boldsymbol{F}(\boldsymbol{x}_k)] \to \min_{\boldsymbol{x}_k^*\in\mathcal{K}} {\boldsymbol{\lambda}_k^*}^\top \boldsymbol{F}(\boldsymbol{x}_k^*)$, suggesting that the expected function value $\mathbb{E}[\boldsymbol{F}(\boldsymbol{x}_k)]$ are optimal. Thus, Definition 3 is indeed derived from Pareto optimality and criticality.

D Missing Proofs for the Non-convergence of MOGA

In this section, we give detailed proof for the non-convergence for MGDA, PCGrad, CAGrad with the example introduced in Section 4. We then prove the non-Lipschitz of the composite weight for the above MOGA algorithms.

D.1 Proof of Theorem 1

Proof. Recall the two-dimensional stochastic optimization setting over the domain $\mathcal{K} = \{(x, y) \mid x \in [-1, 1], y \geq 0\}$

$$\tilde{f}^1(x, y) = \begin{cases} (x+1)^2 + (y-2)^2 & p = 1/2 \\ (x+5)^2 + (y+2)^2 & p = 1/2 \end{cases}, \tilde{f}^2(x, y) = \begin{cases} (x-1)^2 + (y-2)^2 & p = 1/2 \\ (x-5)^2 + (y+2)^2 & p = 1/2 \end{cases}.$$

The expected function is $f^1(x,y) = (x+3)^2 + y^2 + 8, f^2(x,y) = (x-3)^2 + y^2 + 8$. Therefore, the optimization goal is to minimize the distance towards $(-3,0)$ and $(3,0)$ simultaneously. We can easily know that the Pareto set is the line segment $\{(a,0) \mid a \in [-1,1]\}$.

We can calculate the stochastic gradient in the above setting as

$$\nabla \tilde{f}^1(x,y) = \begin{cases} 2(x+1, y-2) & p = 1/2 \\ 2(x+5, y+2) & p = 1/2 \end{cases}, \nabla \tilde{f}^2(x,y) = \begin{cases} 2(x-1, y-2) & p = 1/2 \\ 2(x-5, y+2) & p = 1/2 \end{cases}.$$

Next, we prove that for all point in domain $\{(a,b) | a \in [-1,1], b \in [0, 0.05]\}$, the expected composite directions $\mathbb{E}[\boldsymbol{d}_k] = (d_k^1, d_k^2)$ of MGDA, PCGrad, CAGrad have positive value in y-axis, i.e. $d_k^2 > 0$, which forces the algorithms to move away from the Pareto set.

Lemma 3 (Sener et al. [40])**.** *The min-norm solver has closed form solution for two-objective case. Specifically, for $\boldsymbol{g}^1, \boldsymbol{g}^2$, the composite weight λ for $\boldsymbol{g}^1$ can be calculated as*

$$\arg\min_{\lambda \in [0,1]} \|\lambda \boldsymbol{g}^1 + (1-\lambda)\boldsymbol{g}^2\| = \min\left\{ \left[\frac{(\boldsymbol{g}^2 - \boldsymbol{g}^1)^\top \boldsymbol{g}^2}{\|\boldsymbol{g}^2 - \boldsymbol{g}^1\|_2^2} \right]_+, 1 \right\}, \tag{5}$$

where the operator $[a]_+ = \max\{a, 0\}$.

Lemma 4. *For any point in domain $\{(a,b) | a \in [-1,1], b \in [0, 0.05]\}$, MGDA has a expected composite directions $\mathbb{E}[\boldsymbol{d}_k] = (d_k^1, d_k^2)$ such that $d_k^2 > 0$.*

Proof. We adopt simple enumeration method to calculate the expected direction.

Case 1: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

From 5, we can get $\lambda = (1-a)/2$, then the direction for (a,b) is

$$\boldsymbol{d}_k = -2(0, b-2).$$

Case 2: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$

From 5, we can get $\lambda = (5-a)/10$, then the direction for (a,b) is

$$\boldsymbol{d}_k = -2(0, b+2).$$

Case 3: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$

From 5, we can get $\lambda = (-3a + 2b + 19)/26$, then the direction for (a,b) is

$$\boldsymbol{d}_k = -2(\frac{4a+6b-8}{13}, \frac{6a+9b-12}{13}).$$

Case 4: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = -2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

From 5, we can get $\lambda = (-3a - 2b + 7)/26$, then the direction for (a,b) is

$$\boldsymbol{d}_k = 2(\frac{4a-6b+8}{13}, \frac{-6a+9b-12}{13}).$$

By summing up the above cases and multiply the probability, we have

$$\mathbb{E}[\boldsymbol{d}_k] = (-\frac{4}{13a}, \frac{12-22b}{13}).$$

Since we know that $b \leq 0.05$, we then know $d_k^2 \geq \frac{12-22*0.05}{13} = \frac{10.9}{13} > 0$. We thus prove the Lemma. □

Lemma 5. *For the point in domain* $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$*, PCGrad has a expected composite directions* $\mathbb{E}[\boldsymbol{d}_k] = (d_k^1, d_k^2)$ *such that* $d_k^2 > 0$*.*

Proof. Similar with the proof of MGDA, we also adopt simple enumeration method to calculate the expected direction.

Case 1: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients has not conflicting, then we know the y-axis value of the composite gradient for (a,b) is $4-2b$.

Case 2: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is

$$-4(b+2) + 8(a^2 + (b+2)^2 - 25)(\frac{b+2}{(a+5)^2 + (b+2)^2} - \frac{b+2}{(a-5)^2 + (b+2)^2}).$$

Case 3: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is

$$-4b + 8(a^2 - 4a + b^2 - 9)(\frac{b-2}{(a+1)^2 + (b-2)^2} - \frac{b+2}{(a-5)^2 + (b+2)^2}).$$

Case 4: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is

$$-4b + 8(a^2 + 4a + b^2 - 9)(\frac{b+2}{(a+5)^2 + (b+2)^2} - \frac{b-2}{(a-1)^2 + (b-2)^2}).$$

Sum up the above, it is easy to know d_k^2 is a increasing function for b, and is creasing for $a \in [-1,0]$ and decreasing for $a \in [0,1]$. Plug in $a = \pm 1, b = 0.05$, we can get $d_k^2 \geq 0.2 > 0$. □

Lemma 6. *For the point in domain* $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$*, CAGrad with* $c \in [0.8,1]$ *has a expected composite directions* $\mathbb{E}[\boldsymbol{d}_k] = (d_k^1, d_k^2)$ *such that* $d_k^2 > 0$*.*

Proof. Similar with the proof of MGDA, we also adopt simple enumeration method to calculate the expected direction.

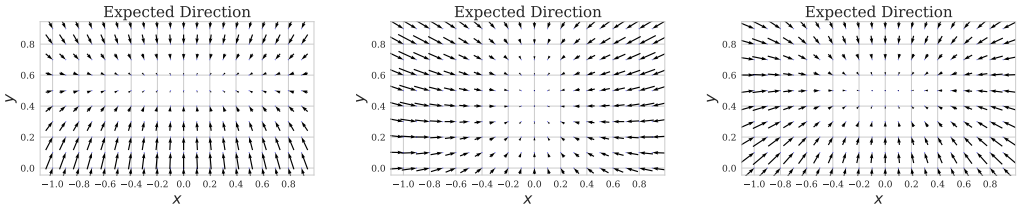
Case 1: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients has not conflicting, then we know the y-axis value of the composite gradient for (a,b) is $-2b+4+2c\sqrt{a^2+(b-2)^2} \geq -2*0.05+4+2c\sqrt{0^2+(0.05-2)^2} = 3.8c+3.9 \geq 6.94$.

Case 2: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$



(a) MGDA (b) PCGrad (c) CAGrad

Figure 3: Expected direction vector field for MGDA, PCGrad, CAGrad. We can observe that this biased direction can move algorithms away from the Pareto optimal set, which resists the algorithm to converge. Additionally, the stationary points for all three algorithms does not lies near the Pareto set.

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is $-2b-4-2c\sqrt{a^2+(b+2)^2} \geq -2*0.05-4-2c\sqrt{1^2+(0+2)^2} = -4.1-c2\sqrt{5} \geq -9$.

Case 3: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+1, b-2), \quad \nabla \tilde{f}^2(a,b) = 2(a-5, b+2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is $-2b+\frac{c\sqrt{2}}{2}\sqrt{(a-4)^2+b^2} \geq -2*0.05+\frac{c\sqrt{2}}{2}\sqrt{(1-4)^2+0^2} = -0.1+3c \leq 2.3$.

Case 4: when the received gradients are

$$\nabla \tilde{f}^1(a,b) = 2(a+5, b+2), \quad \nabla \tilde{f}^2(a,b) = 2(a-1, b-2).$$

It is easy to verify that for $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$, the two gradients consistently conflict with each other, then we know the y-axis value of the composite gradient for (a,b) is $-2b+\frac{c\sqrt{2}}{2}\sqrt{(a+4)^2+b^2} \geq -2*0.05+\frac{c\sqrt{2}}{2}\sqrt{(-1+4)^2+0^2} = -0.1+3c \leq 2.3$.

Sum up the above, we can get $d_k^2 \geq 2.5 > 0$. □

Figure 3 shows the vector field for the composite direction calculated by MGDA, PCGrad, CAGrad, which aligns with the above Lemmas that indicate the expected direction is inverse to the Pareto set. This problem implies that $\mathbb{E}[\boldsymbol{x}_{k+1}]$ is more away from Pareto optimal than $\mathbb{E}[\boldsymbol{x}_k]$ if $\boldsymbol{x}_k$ lies in domain $\{(a,b)|a \in [-1,1], b \in [0,0.05]\}$. Thus the expected suboptimality gap is always > 0, which implies that MGDA, PCGrad, CAGrad do not converge to the optimal solution. In addition, we could also infer from the figure that, the three MOGA algorithms are stationary around $(0, 0.4\ 0.6)$, which represents a large expected suboptimality gap in this case.

□

D.2 Proof of Proposition 2

Proof. The proposition can be proved by the following simple 2 objectives case, where we are provided with $\boldsymbol{g}^1 = (1,0), \boldsymbol{g}^2 = (1-\varepsilon, 0), \boldsymbol{g'}^2 = (1+\varepsilon, 0)$ where $\varepsilon << 1$. For this function, the min-norm direction is $\boldsymbol{g}^2$ for the first case, while is $\boldsymbol{g}^1$ for the second case, which means $\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) = (0,1)$ and $\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2) = (1,0)$. However, we know that $\boldsymbol{g}^2, \boldsymbol{g'}^2$ only have 2ε gap. When ε limits to 0, there exists no finite β to bound $\|\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) - \boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2)\| / \|\boldsymbol{g}^2 - \boldsymbol{g'}^2\|$. □

Generalize to PCGrad. This property can be generalized to PCGrad, we next provide another counterexample to prove it.

Proof. Suppose we are provided with $\boldsymbol{g}^1 = (1,0), \boldsymbol{g}^2 = (-\varepsilon, 0), \boldsymbol{g'}^2 = (\varepsilon, 0), \varepsilon << 1$. For this function, $\boldsymbol{g}^1$ conflicts with $\boldsymbol{g}^2$ while aligned with $\boldsymbol{g'}^2$. This makes $\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) = (\frac{1}{2}+\frac{\varepsilon}{2}, \frac{1}{2}+\frac{1}{2\varepsilon})$ and

$\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2) = (\frac{1}{2}, \frac{1}{2})$. However, we know that $\boldsymbol{g}^2, \boldsymbol{g'}^2$ only have 2ε gap. When ε limits to 0, there exists no finite β to bound $\|\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) - \boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2)\|/\|\boldsymbol{g}^2 - \boldsymbol{g'}^2\| = \sqrt{1 + \frac{1}{\varepsilon^4}}/4$. □

Generalize to CAGrad. This property can also be generalized to CAGrad with parameter $c \in (0, 1/4)$ with the similar example with PCGrad. When $c \in [1/4, +\infty)$, CAGrad performs more like MGDA, it is not difficult to generalize the example of MGDA.

Proof. The proposition can be proved by the following simple 2 objectives case, where we are provided with $\boldsymbol{g}^1 = (1, 0), \boldsymbol{g}^2 = (-\varepsilon, 0), \boldsymbol{g'}^2 = (\varepsilon, 0), \varepsilon << 1$. For this function, $\boldsymbol{g}^1$ conflicts with $\boldsymbol{g}^2$ while aligned with $\boldsymbol{g'}^2$. This makes $\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) = (\frac{1-c(1-\varepsilon)}{2}, \frac{1}{2})$ and $\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2) = (\frac{1+c(1+\varepsilon)}{2}, \frac{1}{2})$. However, we know that $\boldsymbol{g}^2, \boldsymbol{g'}^2$ only have 2ε gap. When ε limits to 0, there exists no finite β to bound $\|\boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g}^2) - \boldsymbol{\lambda}(\boldsymbol{g}^1, \boldsymbol{g'}^2)\|/\|\boldsymbol{g}^2 - \boldsymbol{g'}^2\| = c/\varepsilon$. □

E Missing Proofs for the Convergence of CRMOGA

We here give a rigorous convergence analysis of CRMOGA in the stochastic (mini-batch) gradient setting. We first fulfill the proofs for Lemma 1 and Lemma 2. Before providing the final proof, we would like to introduce some useful and insightful lemmas that reflect the algorithmic property and can be used for the final proof. Specifically, Appendix E.4 and E.5 decompose the convex and non-convex convergence bounds into sub-terms bounded by lemmas from Appendix E.3.

E.1 Proof of Lemma 1

Lemma 1. *Under Assumption 1, 2. The correlation between $\boldsymbol{\lambda}_k$ and $\boldsymbol{G}_k$ can be bounded as $\|\mathbb{E}[\boldsymbol{G}_k\boldsymbol{\lambda}_k] - \mathbb{E}[\boldsymbol{G}_k]\mathbb{E}[\boldsymbol{\lambda}_k]\|_2^2 \leq m\sigma^2\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k)$, where $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) = \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{\lambda}_k - \mathbb{E}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k]\|_2^2]$.*

Proof. Denote $\boldsymbol{g}^i(\boldsymbol{x}_k, \boldsymbol{\xi}_k^i) = \boldsymbol{g}^i(\boldsymbol{x}_k), i \in [m]$ for simplicity. By the unbiasness of each stochastic gradient $\mathbb{E}[\boldsymbol{g}^i(\boldsymbol{x}_k)] = \nabla f^i(\boldsymbol{x}_k)$ for each objective, we have

$$\begin{aligned}
\|\mathbb{E}[G_k\boldsymbol{\lambda}_k] - \mathbb{E}[G_k]\mathbb{E}[\boldsymbol{\lambda}_k]\|_2^2 &= \left\|\mathbb{E}[\sum_{i=1}^m \boldsymbol{\lambda}_k^i \boldsymbol{g}^i(\boldsymbol{x}_k)] - \mathbb{E}[\sum_{i=1}^m \boldsymbol{\lambda}_k^i \nabla f^i(\boldsymbol{x}_k)]\right\|_2^2 \\
&= \left\|\mathbb{E}[\sum_{i=1}^m \boldsymbol{\lambda}_k^i (\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))]\right\|_2^2 \\
&= \left\|\mathbb{E}[\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k]^i)(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))]\right\|_2^2.
\end{aligned}$$

The last equation is by the fact that

$$\mathbb{E}[\sum_{i=1}^m \mathbb{E}[\lambda_k]^i(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))] = \sum_{i=1}^m \mathbb{E}[\lambda_k]^i \mathbb{E}[\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)] = 0,$$

where the first equation is by the linearity of expectation, and the second one is by the unbiasness of each stochastic gradient. Further, by the fact that $\|\mathbb{E}[a]\| \leq \mathbb{E}[\|a\|]$ and the triangle inequality for the l_2 norm, we know that

$$\begin{aligned}
\|\mathbb{E}[\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k]^i)(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))]\|_2^2 &\leq \mathbb{E}[\|\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k]^i)(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\|]^2 \\
&\leq (\mathbb{E}[\sum_{i=1}^m \|(\lambda_k^i - \mathbb{E}[\lambda_k]^i)(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\|])^2.
\end{aligned}$$

Where, since $2ab \leq \beta a^2 + \frac{1}{\beta}b^2$ for $\beta > 0$, we can get

$$\begin{aligned}\|(\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\| &= |\lambda_k^i - \mathbb{E}[\lambda_k^i]|\|(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\| \\ &\leq \frac{\beta_k}{2}(\lambda_k^i - \mathbb{E}[\lambda_k^i])^2 + \frac{1}{2\beta_k}\|(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\|^2\end{aligned}$$

Plug in, we obtain

$$\begin{aligned}&(\mathbb{E}[\sum_{i=1}^m \|(\lambda_k^i - \mathbb{E}[\lambda_k]^i)(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\|])^2 \\ &\leq \mathbb{E}\left[\sum_{i=1}^m \frac{\beta_k}{2}(\lambda_k^i - \mathbb{E}[\lambda_k]^i)^2 + \sum_{i=1}^m \frac{\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\|^2}{2\beta_k}\right]^2 \\ &= (\frac{\beta_k}{2}\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k) + \frac{\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))}{2\beta_k})^2.\end{aligned}$$

The last equation is by the definition of $\mathbb{V}_{\boldsymbol{\xi}_k}$. Here we let $\beta_k = \frac{\sqrt{\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))}}{\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)}}$, and apply all the above result, we finally get

$$\|\mathbb{E}[G_k\lambda_k] - \mathbb{E}[G_k]\mathbb{E}[\lambda_k]\|_2^2 \leq \mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)(\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))) \leq m\sigma^2\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k).$$

We thus prove the Lemma. □

Remark. Lemma 1 suggests that the correlation between the composite weights and stochastic gradients is bounded by both the variance of composite weights $\boldsymbol{\lambda}_k$ and multiple stochastic gradients $\boldsymbol{g}^i(\boldsymbol{x}_k)$. Therefore, there are two effective ways to decrease this correlation: reduce the variance of λ_k, or the variance of $\boldsymbol{g}^i(\boldsymbol{x}_k)$. For the latter way, one can adopt variance reduction framework [20, 4] for stochastic optimization. However, it [20] requires global gradients that may be computational expensive, or slows down the convergence rate [4]. For the former one, the momentum mechanism on $\boldsymbol{\lambda}_k$ is a simple and effective way, which enhances both theoretical and empirical results without additional computational cost.

E.2 Proof of Lemma 2

Lemma 2. *Under Assumption 1, $\boldsymbol{\lambda}_k$ variance for Algorithm 1 is bounded $\mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k] \leq m^2B^2(1-\alpha_k)^2$.*

Proof. By the definition of Algorithm 1, we know that

$$\boldsymbol{\lambda}_k = \alpha_k\boldsymbol{\lambda}_{k-1} + (1-\alpha_k)\hat{\boldsymbol{\lambda}}_k.$$

From the fact that $\mathbb{E}[\|X - \mathbb{E}[X]\|_2^2] \leq \mathbb{E}[\|X - a\|_2^2]$ for any a, we can get

$$\begin{aligned}\mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k] &= \mathbb{E}_{\boldsymbol{\xi}_k}[\|\lambda_k - \mathbb{E}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k]\|_2^2] \leq \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1}\|_2^2] \\ &= \mathbb{E}_{\boldsymbol{\xi}_k}[\|(1-\alpha_k)(\hat{\boldsymbol{\lambda}} - \lambda_{k-1})\|_2^2] = (1-\alpha_k)^2\mathbb{E}_{\boldsymbol{\xi}_k}[\|(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}_{k-1})\|_2^2] \leq m^2B^2(1-\alpha_k)^2.\end{aligned}$$

The last inequality is by the fact $0 \leq \lambda_k^i, \hat{\lambda}_k^i \leq B, i = 1, \ldots, m$. □

E.3 Lemmas needed for Proving Thm2/Thm3

This subsection presents several useful Lemmas for proving Theorem 2 and Theorem 3. Lemma 1 measures the expected gap between $-\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)$ and d_k. The following Lemma bounds the dot production between them, which measures the local decrease of the composite functional value.

Lemma 7. *Under Assumption 1, 2, 3, for any gradient descent algorithm updated with composite gradient $\boldsymbol{G}_k\boldsymbol{\lambda}_k$, we have the following inequality*

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right] \leq 2mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} - \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right].$$

Proof. We first decompose the term into

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right] = \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right] - \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right].$$

The first term can potentially corrupt the effectiveness of optimization, and the second term measures the descent value. We next bound the first term. By definitions and decomposition, we get

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\sum_{i=1}^m \lambda_k^i (\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \text{ (term A)} \\
&\quad + \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i]) \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \text{ (term B)} \\
&\quad + \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i] \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \text{ (term C).}
\end{aligned}$$

We then bound each term individually. By Assumption 1 and Assumption 3, we know that $\lambda_k^i \in [0, B], \|\nabla f^i(\boldsymbol{x}_k)\| \leq H, \forall i = 1, \ldots, m$. By Cauchy–Schwartz inequality, we further know that for the term A

$$\begin{aligned}
\text{term A} &= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2 \left\|\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \left\|\nabla f^i(\boldsymbol{x}_k)\right\|_2\right) \left\|\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m BH\right) \left\|\sum_{i=1}^m (\lambda_k^i - \mathbb{E}[\lambda_k^i])(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right] \\
&\leq mBH\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m |\lambda_k^i - \mathbb{E}[\lambda_k^i]| \|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\|_2\right].
\end{aligned}$$

Similar with the proof in Lemma 1, by the fact that $ab \leq \frac{1}{2\beta_k}a^2 + \frac{\beta_k}{2}b^2$ for any $\beta_k > 0$, and by the linearity of expectation, we can get

$$\begin{aligned}
\text{term A} &\leq \frac{mBH}{2\beta_k}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\xi}_k}\left[|\lambda_k^i - \mathbb{E}[\lambda_k^i]|^2\right] + \frac{mBH\beta_k}{2}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\|_2^2\right] \\
&\leq \frac{mBH}{2\beta_k}\mathbb{V}[\boldsymbol{\lambda}_k] + \frac{mBH\beta_k}{2}\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)].
\end{aligned}$$

By setting $\beta_k = \sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]}/\sqrt{\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}$, we have

$$\text{term A} \leq mBH\sqrt{\mathbb{V}[\boldsymbol{\lambda}_k]\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

With similar tricks, we have for the term B

$$\begin{aligned}
\text{term B} &= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m(\lambda_k^i - \mathbb{E}[\lambda_k^i])\nabla f^i(\boldsymbol{x}_k)\right)^\top\left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m(\lambda_k^i - \mathbb{E}[\lambda_k^i])\nabla f^i(\boldsymbol{x}_k)\right\|_2\left\|\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m H|\lambda_k^i - \mathbb{E}[\lambda_k^i]|\right)\left\|\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m H|\lambda_k^i - \mathbb{E}[\lambda_k^i]|\left\|\sum_{j=1}^m \mathbb{E}[\lambda_k^j](\nabla f^j(\boldsymbol{x}_k) - \boldsymbol{g}^j(\boldsymbol{x}_k))\right\|_2\right] \\
&\leq H\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \frac{1}{2\beta_k}|\lambda_k^i - \mathbb{E}[\lambda_k^i]|^2 + \frac{\beta_k}{2}\left\|\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2^2\right)\right] \\
&\leq H\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \frac{1}{2\beta_k}|\lambda_k^i - \mathbb{E}[\lambda_k^i]|^2 + \frac{\beta_k}{2}(\sum_{i=1}^m \mathbb{E}[\lambda_k^i]^2)(\sum_{i=1}^m \left\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\right\|_2^2)\right)\right] \\
&\leq \frac{H}{2\beta_k}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\xi}_k}\left[|\lambda_k^i - \mathbb{E}[\lambda_k^i]|^2\right] + \frac{mB^2H\beta_k}{2}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\|_2^2\right] \\
&= \frac{H}{2\beta_k}\mathbb{V}[\boldsymbol{\lambda}_k] + \frac{mB^2H\beta_k}{2}\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)].
\end{aligned}$$

The first inequality is by Cauchy–Schwarz inequality. The second one is by the triangle inequality of l_2 norm, we have $\|\sum_{i=1}^m(\lambda_k^i - \mathbb{E}[\lambda_k^i])\nabla f^i(\boldsymbol{x}_k)\|_2 \leq \sum_{i=1}^m \|(\lambda_k^i - \mathbb{E}[\lambda_k^i])\nabla f^i(\boldsymbol{x}_k)\|_2 = \sum_{i=1}^m |\lambda_k^i - \mathbb{E}[\lambda_k^i]|\|\nabla f^i(\boldsymbol{x}_k)\|_2$. With Assumption 3, we know $\sum_{i=1}^m |\lambda_k^i - \mathbb{E}[\lambda_k^i]|\|\nabla f^i(\boldsymbol{x}_k)\|_2 \leq H\sum_{i=1}^m |\lambda_k^i - \mathbb{E}[\lambda_k^i]|$. The third one is from the fact that $ab \leq \frac{1}{2\beta_k}a^2 + \frac{\beta_k}{2}b^2$. The forth one is also from Cauchy-Schwartz inequality. The last one is from Assumption 1 that states $\mathbb{E}[\lambda_k^i] \leq B$. Further by setting $\beta_k = \sqrt{\mathbb{V}[\boldsymbol{\lambda}_k]}/B\sqrt{\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}$, we have

$$\text{term B} \leq \frac{(m+1)BH}{2}\sqrt{\mathbb{V}[\boldsymbol{\lambda}_k]\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} \leq mBH\sqrt{\mathbb{V}[\boldsymbol{\lambda}_k]\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

For term C, by the fact that only $\boldsymbol{g}^i(\boldsymbol{x}_k)$ has randomness, we get

$$\begin{aligned}
\text{term C} &= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i]\nabla f^i(\boldsymbol{x}_k)\right)^\top\left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i](\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right)\right] \\
&= \left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i]\nabla f^i(\boldsymbol{x}_k)\right)^\top\left(\sum_{i=1}^m \mathbb{E}[\lambda_k^i]\left(\nabla f^i(\boldsymbol{x}_k) - \mathbb{E}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]\right)\right) = 0.
\end{aligned}$$

By summing up the above results, we obtain

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top\left(\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)\right] \leq 2mBH\sqrt{\mathbb{V}[\boldsymbol{\lambda}_k]\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

Plug in the first decomposition in the beginning, we finally get

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right] \leq 2mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} - \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right].$$

□

Remark. The result from Lemma 7 indicates that the local decrease for expected composite loss $\mathbb{E}[\boldsymbol{\lambda}_k]^T \boldsymbol{F}(\boldsymbol{x})$ is bounded by $2mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} - \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right]$. The first term consists of the variances for composite weights and multiple gradients, and the second term is the negative norm of the effective direction. When the solution approximates the optimal, the second term would be smaller than the former term when the variances are large, then the value of the composite loss will increase, leading to the non-convergence. This result is aligned with the non-convergence for MOGA algorithms.

We next give a second-order bias analysis of the composite gradient $\boldsymbol{d}_k$ by the following lemma.

Lemma 8. *Under Assumption 1, 2, 3, for any gradient descent algorithm updated with composite gradient $\boldsymbol{G}_k\boldsymbol{\lambda}_k$, we have the following inequality*

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2 - \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \leq mB^2\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i) + 4mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

Proof. We first decompose the expectation into

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2 - \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2 - \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \\
&= \underbrace{\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right]}_{\text{term A}} - \underbrace{2\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right]}_{\text{term B}}.
\end{aligned}$$

We next analyze the term A. By the triangle inequality of the l_2 norm, we have

$$\begin{aligned}
\text{term A} &= \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right\|_2^2\right] = \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i (\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2^2\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \left\|(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|_2\right)^2\right],
\end{aligned}$$

Further, by the fact that $\lambda_k^i \in [0, B], i = 1, \ldots, m$, we know that

$$\begin{aligned}
\text{term A} &\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m {\lambda_k^i}^2\right)\left(\sum_{i=1}^m \left\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\right\|_2^2\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[mB^2\sum_{i=1}^m \left\|(\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k))\right\|^2\right] = mB^2\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).
\end{aligned}$$

We then analyze the term B. Similar with the proof in Lemma 7, and by the fact that the minus sign does not affect the inequality, we know that

$$\text{term B} = -2\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \left(\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right]$$
$$\leq 4mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

Combining all the results, we obtain

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2 - \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \leq mB^2\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i) + 4mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}.$$

□

Remark. In the single-objective stochastic optimization, the second term is zero, leaving only the gradient variance in the bound. While in the multi-objective stochastic optimization, the second term is non-zero and related to the number of objectives, the variance of composite weight, and the gradient variance, which makes it significantly different.

E.4 Proof for Theorem 2 (Convex Convergence)

Theorem 2. *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x})$ are all convex with bounded optimal values as $f^1(\boldsymbol{x}^*), \ldots, f^m(\boldsymbol{x}^*) \leq F$ for $\boldsymbol{x}^*$ in Pareto set, and the distance from sequence to Pareto set is bounded, i.e., $\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq D$. Set $0 \leq \eta_n \leq \ldots \leq \eta_1 \leq 1/mLB, \alpha_k \in (0,1]$ for $k = 1, \ldots, n$ in Algorithm 1, it achieves*

$$\frac{1}{n}\sum_{k=1}^n \mathbb{E}\left[\max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} (\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right] \leq \frac{D^2}{\eta_n n} + \frac{m^2B^2(\sigma^2 + \sigma H + H^2)}{n}\sum_{k=1}^n \eta_k$$
$$+ \frac{DB\sigma m^{3/2}}{n}\sum_{k=1}^n (1-\alpha_k) + \frac{6m^{5/2}B^2H\sigma}{n}\sum_{k=1}^n (1-\alpha_k)\eta_k + \frac{2F}{n}\sum_{l=1}^n l(1-\alpha_{l+1})\sum_{i=1}^m |\lambda_l^i - \hat{\lambda}_{l+1}^i|.$$

To prove Theorem 2, it is sufficient to prove the bound for

$$\frac{1}{n}\sum_{k=1}^n \mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \min_{\boldsymbol{x}_k^* \in \mathcal{K}} \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)]. \tag{6}$$

We next provide the reason. Specifically, from Assumption 1, we have $\sum_{i=1}^m \lambda_k^i \geq 1$ for $k = 1, \ldots, n$. Therefore, we know that

$$\frac{1}{n}\sum_{k=1}^n \mathbb{E}[\frac{\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})}{\sum_{j=1}^m \lambda_k^j} - \frac{\min_{\boldsymbol{x}_k^* \in \mathcal{K}} \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)}{\sum_{i=1}^m \lambda_k^i}] \leq \frac{1}{n}\sum_{k=1}^n \mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \min_{\boldsymbol{x}_k^* \in \mathcal{K}} \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)].$$

Furthermore, denote $\bar{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k / \sum_{j=1}^m \lambda_k^j$. By the property for the max-min operator, the left side of the above inequality can upper bound what we need to bound in Theorem 2 as

$$\frac{1}{n}\sum_{k=1}^n \mathbb{E}\left[\max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} (\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right] \leq \frac{1}{n}\sum_{k=1}^n \mathbb{E}[\sum_{i=1}^m \bar{\boldsymbol{\lambda}}_k^\top F(\boldsymbol{x}_k) - \min_{\boldsymbol{x}_k^* \in \mathcal{K}} \bar{\boldsymbol{\lambda}}_k^\top F(\boldsymbol{x}_k^*)].$$

Therefore, the rest of the proof is to bound 6. Before the final proof, we first introduce several Lemmas. The following Lemma give an induction as

Lemma 9. *Under the same assumption as Theorem 2, for any gradient descent algorithm updated with composite gradient $\boldsymbol{G}_k\boldsymbol{\lambda}_k$, and any $\boldsymbol{x}^*$ in Pareto set, we have the following inequality*

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)] \leq \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right]$$
$$+ D\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)\left(\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))\right)} + 6\eta_k mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}$$
$$+ \eta_k mB^2\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i) + \eta_k mB^2 H\left(mH + \sqrt{m\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k]}\right).$$

Proof. From the L-smoothness of each objective function, we have

$$\lambda_k^i f^i(\boldsymbol{x}_{k+1}) \leq \lambda_k^i\left(f^i(\boldsymbol{x}_k) + \eta_k\nabla f^i(\boldsymbol{x}_k)^\top(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{L}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2\right).$$

By the iterative rule, we know that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k\boldsymbol{d}_k$. Further, by the convexity of f^i, and denote $\boldsymbol{x}^*$ as any point in the feasible set, we have $f^i(\boldsymbol{x}_k) \leq f^i(\boldsymbol{x}^*) + \nabla f^i(\boldsymbol{x}_k)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*)$. Then we can get

$$\lambda_k^i f^i(\boldsymbol{x}_{k+1}) \leq \lambda_k^i f^i(\boldsymbol{x}_k) + \eta_k\lambda_k^i\nabla f^i(\boldsymbol{x}_k)^\top\boldsymbol{d}_k + \frac{L\eta_k^2}{2}\lambda_k^i\|\boldsymbol{d}_k\|_2^2$$
$$\leq \lambda_k^i f^i(\boldsymbol{x}^*) + \lambda_k^i\nabla f^i(\boldsymbol{x}_k)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*) + \eta_k\lambda_k^i\nabla f^i(\boldsymbol{x}_k)^\top\boldsymbol{d}_k + \frac{L\eta_k^2}{2}\lambda_k^i\|\boldsymbol{d}_k\|_2^2.$$

Summing up above for $i = 1, \ldots, m$, we have

$$\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)$$
$$\leq \left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*) + \eta_k\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top\boldsymbol{d}_k + \frac{L\eta_k^2}{2}\left(\sum_{i=1}^m \lambda_k^i\right)\|\boldsymbol{d}_k\|_2^2.$$

By setting $\eta_k \leq 1/mLB$, we can know that $L\eta_k\left(\sum_{i=1}^m \lambda_k^i\right) \leq L\eta_k mB \leq 1$. Then, the third term can be bounded as $\frac{L\eta_k^2}{2}\left(\sum_{i=1}^m \lambda_k^i\right)\|\boldsymbol{d}_k\|_2^2 \leq \frac{\eta_k}{2}\|\boldsymbol{d}_k\|_2^2$. Plugging such a bound for the third term, and take the expectation on random variable $\boldsymbol{\xi}_k$, we then obtain

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)\right]$$
$$\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*) + \eta_k\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top\boldsymbol{d}_k + \frac{\eta_k}{2}\|\boldsymbol{d}_k\|_2^2\right].$$

By the linearity of expectation, we rearrange the above term into

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)]$$
$$\leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*)\right] + \eta_k\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top\boldsymbol{d}_k\right] + \frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right].$$

We next to bound the first term. By rearranging the equation, we get

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*)\right] = \mathbb{E}_{\boldsymbol{\xi}_k}\left[-\boldsymbol{d}_k^\top(\boldsymbol{x}_k - \boldsymbol{x}^*) + \left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*)\right]$$
$$= \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right] + \frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right] + \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i\nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right)^\top(\boldsymbol{x}_k - \boldsymbol{x}^*)\right],$$

where the last equation is by the fact that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \boldsymbol{d}_k$. Since we know that

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right]^\top (\boldsymbol{x}_k - \boldsymbol{x}^*) \leq \left\|\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right]\right\|_2 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$$
$$\leq D\left\|\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right]\right\|_2 = D\sqrt{\left\|\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \boldsymbol{d}_k\right]\right\|_2^2}$$
$$\leq D\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)\left(\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))\right)}.$$

The second inequality is by the assumption that the set $\{\boldsymbol{x}_k, k = 1, \ldots, K\}$ is a subset of a bounded set with diagram D, and the third one is from Lemma 1. By combining all the above results, we have

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)]$$
$$\leq \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right] + D\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)\left(\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))\right)} + \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right]$$
$$+ \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right]$$
$$\leq \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right] + D\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k)}$$
$$+ \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2 - \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] + 2\eta_k mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} \quad \text{(from Lemma 7)}$$
$$\leq \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right] + D\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}(\lambda_k)\left(\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k))\right)}$$
$$+ 6\eta_k mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \eta_k mB^2 \sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i) \qquad \text{(from Lemma 8).}$$

In addition, there is a slight different with the Lemma that we aim to prove: the left side is $\boldsymbol{x}_{k+1}$ not $\boldsymbol{x}_k$. The following we fix this issue. By the H-Lipschitz continuous for each objective, we know that

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})] \leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i H\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|)] \leq mBH\eta_k \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k\|].$$

By the triangle inequality of norm and the definition for the composite gradient $\boldsymbol{d}_k$, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k\|] &= \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k + \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\| + \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|] \\
&= \mathbb{E}_{\boldsymbol{\xi}_k}[\|\sum_{i=1}^m \lambda_k^i (\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\| + \|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\|] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i \|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\| + \sum_{i=1}^m \lambda_k^i \|\nabla f^i(\boldsymbol{x}_k)\|] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i \|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\| + \sum_{i=1}^m \lambda_k^i H] \\
&\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i \|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k)\|] + mBH.
\end{aligned}$$

The third inequality is from the H-Lipschitz, and the last one is from Assumption 1. By the fact that $ab \leq \frac{1}{2\beta_k}a^2 + \frac{\beta_k}{2}b^2$ for any $\beta_k > 0$, we can have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k\|] &\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \frac{1}{2\beta_k}(\lambda_k^i)^2 + \sum_{i=1}^m \frac{\beta_k}{2}\|\nabla f^i(\boldsymbol{x}_k) - \boldsymbol{g}^i(\boldsymbol{x}_k\|^2] + mBH \\
&\leq \frac{mB^2}{2\beta_k} + \frac{\beta_k}{2}\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k] + mBH.
\end{aligned}$$

Let $\beta_k = B\sqrt{m}/\sqrt{\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k]}$, we have $\mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{d}_k\|] \leq B\sqrt{m\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k]} + mBH$. Plug in the result, we can get

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})] \leq \eta_k mB^2H(mH + \sqrt{m\sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k]}).$$

Combining all the above, we prove the lemma. □

By summing up the above lemma, we get the following one

Lemma 10. *Under the same assumption as Theorem 2, for any $\boldsymbol{x}^*$ in Pareto set, we have the following inequality for CRMOGA algorithm*

$$\begin{aligned}
\sum_{k=1}^n \mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}^*)] &\leq \left(\frac{1}{2\eta_n} + \frac{1}{2\eta_1}\right) D^2 + m^{3/2}DB\sigma\sum_{k=1}^n (1-\alpha_k) \\
&+ 6m^{5/2}B^2H\sigma\sum_{k=1}^n (1-\alpha_k)\eta_k + m^2B^2(\sigma^2 + \sigma H + H^2)\sum_{k=1}^n \eta_k.
\end{aligned}$$

Proof of Theorem 2. From Lemma 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k] \leq m^2B^2(1-\alpha_k)^2$. From Assumption 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k)) \leq \sigma^2$. Then plug into Lemma 9, we get

$$\begin{aligned}
\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\xi}_k}[\lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \lambda_k^i f^i(\boldsymbol{x}^*)] &\leq \frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2^2\right] \\
&+ mDB\sigma\sqrt{m}(1-\alpha_k) + 6m^2B^2H\sigma\sqrt{m}(1-\alpha_k)\eta_k + m^2B^2(\sigma^2 + \sigma H + H^2)\eta_k.
\end{aligned}$$

Sum up the above inequality from $k=1$ to $k=n$, we have

$$\sum_{k=1}^{n}\sum_{i=1}^{m}\mathbb{E}_{\boldsymbol{\xi}_k}[\lambda_k^i f^i(\boldsymbol{x}_{k+1})-\lambda_k^i f^i(\boldsymbol{x}^*)] \leq \sum_{k=1}^{n}\frac{1}{2\eta_k}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2^2-\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2^2\right]$$
$$+\sum_{k=1}^{n}(mDB\sigma\sqrt{m}(1-\alpha_k)+6m^2B^2H\sigma\sqrt{m}(1-\alpha_k)\eta_k+m^2B^2(\sigma^2+\sigma H+H^2)\sum_{k=1}^{n}\eta_k).$$

Take expectation of $\xi_1,\xi_2,\ldots,\xi_n$ on the both sides, we can get

$$\sum_{k=1}^{n}\sum_{i=1}^{m}\mathbb{E}[\lambda_k^i f^i(\boldsymbol{x}_{k+1})-\lambda_k^i f^i(\boldsymbol{x}^*)] \leq \sum_{k=1}^{n}\frac{1}{2\eta_k}\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2^2-\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2^2\right]$$
$$+DB\sigma m\sqrt{m}\sum_{k=1}^{n}(1-\alpha_k)+6m^2B^2H\sigma\sqrt{m}\sum_{k=1}^{n}(1-\alpha_k)\eta_k+m^2B^2(\sigma^2+\sigma H+H^2)\sum_{k=1}^{n}\eta_k.$$

By the fact that $\eta_k \geq \eta_{k+1}$, we then bound the first term as

$$\begin{aligned}
&\sum_{k=1}^{n}\frac{1}{2\eta_k}\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2^2-\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2^2\right]\\
&=\sum_{k=2}^{n}(\frac{1}{2\eta_k}-\frac{1}{2\eta_{k-1}})\mathbb{E}\left[\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2^2\right]+\frac{1}{2\eta_1}\mathbb{E}\left[\|\boldsymbol{x}_1-\boldsymbol{x}^*\|_2^2\right]-\frac{1}{2\eta_n}\mathbb{E}\left[\|\boldsymbol{x}_{n+1}-\boldsymbol{x}^*\|_2^2\right]\\
&\leq\sum_{k=2}^{n}(\frac{1}{2\eta_k}-\frac{1}{2\eta_{k-1}})D^2+\frac{1}{2\eta_1}D^2\leq\left(\frac{1}{2\eta_n}+\frac{1}{2\eta_1}\right)D^2.
\end{aligned}$$

Plug in the above result, we have

$$\begin{aligned}
\sum_{k=1}^{n}\mathbb{E}[\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_{k+1})-\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}^*)] &\leq \left(\frac{1}{2\eta_n}+\frac{1}{2\eta_1}\right)D^2+m^{3/2}DB\sigma\sum_{k=1}^{n}(1-\alpha_k)\\
&+6m^{5/2}B^2H\sigma\sum_{k=1}^{n}(1-\alpha_k)\eta_k+m^2B^2(\sigma^2+\sigma H+H^2)\sum_{k=1}^{n}\eta_k.
\end{aligned}$$

We thus prove the lemma. □

By observing Lemma 10 and the final version we need to proof, the only different is the compared optimum is $\boldsymbol{x}^*$ for all k not the adaptive one $\boldsymbol{x}_k^*=\arg\min_{\boldsymbol{x}\in\mathcal{K}}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x})$, the following lemma measures such a gap.

Lemma 11. *Under the same assumption as Theorem 2, for any $\boldsymbol{x}^*$ in Pareto set, we have the following inequality for CRMOGA algorithm*

$$\mathbb{E}\left[\min_{\boldsymbol{x}^*\in\mathcal{K}}\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}^*)-\sum_{k=1}^{n}\min_{\boldsymbol{x}_k^*\in\mathcal{K}}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right]\leq 2F\sum_{l=1}^{n}l(1-\alpha_k)\sum_{i=1}^{m}|\lambda_l^i-\hat{\lambda}_{l+1}^i|$$

Proof. Denote $\bar{\lambda}^i=\sum_{k=1}^{n}\lambda_k^i/n$. Denote $\boldsymbol{x}^*=\arg\min_{\boldsymbol{x}^*\in\mathcal{K}}\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}^*)$ and $\boldsymbol{x}_k^*=\arg\min_{\boldsymbol{x}_k^*\in\mathcal{K}}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*), k=1,\ldots,n$. We first rearrange into

$$\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}^*)-\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right]=\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}\bar{\lambda}^i f^i(\boldsymbol{x}^*)-\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right].$$

Since $\boldsymbol{x}^*$ is the minimum for $\bar{\lambda}_k^i f^i(\boldsymbol{x})$, we then have

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}^*)-\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right] &\leq \mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}\bar{\lambda}^i f^i(\boldsymbol{x}_k^*)-\sum_{k=1}^{n}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right]\\
&=\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}(\bar{\lambda}^i-\lambda_k^i)f^i(\boldsymbol{x}_k^*)\right]\\
&\leq\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}|\bar{\lambda}^i-\lambda_k^i||f^i(\boldsymbol{x}_k^*)|\right].
\end{aligned}$$

By the assumption that objective functions $f^1(\boldsymbol{x}_k), \ldots, f^m(\boldsymbol{x}_k) \leq F$ are all bounded for $k = 1, \ldots, n$, we get

$$\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}|\bar{\lambda}^i - \lambda_k^i||f^i(\boldsymbol{x}_k^*)|\right] \leq F\mathbb{E}\left[\sum_{k=1}^{n}\sum_{i=1}^{m}|\bar{\lambda}^i - \lambda_k^i|\right].$$

As we know that

$$\begin{aligned}
\sum_{k=1}^{n}\sum_{i=1}^{m}|\bar{\lambda}^i - \lambda_k^i| &= \sum_{k=1}^{n}\sum_{i=1}^{m}|\frac{1}{n}\sum_{t=1}^{n}(\lambda_t^i - \lambda_k^i)| = \frac{1}{n}\sum_{k=1}^{n}\sum_{i=1}^{m}|\sum_{t=1}^{n}(\lambda_t^i - \lambda_k^i)| \\
&\leq \frac{1}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{t=1}^{n}|\lambda_t^i - \lambda_k^i| = \frac{1}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{t=1}^{k-1}|\lambda_t^i - \lambda_k^i| + \frac{1}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{t=k+1}^{n}|\lambda_t^i - \lambda_k^i| \\
&\leq \frac{2}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{t=k}^{n}\sum_{l=k}^{t}|\lambda_l^i - \lambda_{l+1}^i| = \frac{2}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{l=k}^{n}\sum_{t=l}^{n}|\lambda_l^i - \lambda_{l+1}^i| \\
&= \frac{2}{n}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{l=k}^{n}(n-l+1)|\lambda_l^i - \lambda_{l+1}^i| \leq 2\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{l=k}^{n}|\lambda_l^i - \lambda_{l+1}^i| \\
&= 2\sum_{i=1}^{m}\sum_{l=1}^{n}\sum_{k=1}^{l}|\lambda_l^i - \lambda_{l+1}^i| = 2\sum_{i=1}^{m}\sum_{l=1}^{n}l|\lambda_l^i - \lambda_{l+1}^i| \\
&= 2\sum_{i=1}^{m}\sum_{l=1}^{n}l(1-\alpha_k)|\lambda_l^i - \hat{\lambda}_{l+1}^i| = 2\sum_{l=1}^{n}l(1-\alpha_{l+1})\sum_{i=1}^{m}|\lambda_l^i - \hat{\lambda}_{l+1}^i|.
\end{aligned}$$

The last two equation is by the definition of Algorithm 1 □

We are now ready to proof Theorem 2. Combine Lemma 10 and Lemma 11, we have

$$\begin{aligned}
&\sum_{k=1}^{n}\mathbb{E}[\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \min_{\boldsymbol{x}_k^*\in\mathcal{K}}\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)] \leq \left(\frac{1}{2\eta_n} + \frac{1}{2\eta_1}\right)D^2 + DB\sigma m^{3/2}\sum_{k=1}^{n}(1-\alpha_k) \\
&+ 6m^{5/2}B^2H\sigma\sum_{k=1}^{n}(1-\alpha_k)\eta_k + m^2B^2(\sigma^2 + \sigma H + H^2)\sum_{k=1}^{n}\eta_k + 2F\sum_{l=1}^{n}l(1-\alpha_{l+1})\sum_{i=1}^{m}|\lambda_l^i - \hat{\lambda}_{l+1}^i|.
\end{aligned}$$

By averaging the above, and the fact that $\eta_n \leq \eta_1$, we prove the theorem.

E.5 Proof for Theorem 3 (Non-convex Convergence)

Theorem 3. *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}_k), \ldots, f^m(\boldsymbol{x}_k) \leq F$ are all bounded for $k = 1, \ldots, n$. Set $0 \leq \eta_n \leq \ldots \leq \eta_1 \leq 1/mLB$ and $\alpha_k \in (0, 1]$ in Algorithm 1, it achieves*

$$\begin{aligned}
\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^*\in S_m}\|\nabla\boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right] \leq \frac{F}{n}\sum_{k=2}^{n}\frac{2}{\eta_k}(1-\alpha_k)\sum_{i=1}^{m}|\hat{\lambda}_k^i - \lambda_{k-1}^i| + \frac{2mBF}{n\eta_n} \\
+ \frac{4m^{5/2}B^2H\sigma}{n}\sum_{k=1}^{n}(1-\alpha_k) + \frac{Lm^3B^3\sigma^2}{2n}\sum_{k=1}^{n}\eta_k.
\end{aligned}$$

Similar to the convex case, to prove Theorem 3, it is sufficient to prove the bound for

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|\nabla\boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k\|^2\right], \tag{7}$$

where $\boldsymbol{\lambda}_k$ is generated by MOGA or CRMOGA algorithms. The reason is also similar to the convex case, which is that 7 upper bounds the left side in Theorem 3 because

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k\|^2\right] \geq \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\frac{\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k\|^2}{(\sum_{i=1}^m \lambda_k^i)^2}\right] \\
&= \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\bar{\boldsymbol{\lambda}}_k\|^2\right] \geq \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^* \in S_m}\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right],
\end{aligned}$$

where $\bar{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k/(\sum_{i=1}^m \lambda_k^i)^2 \in S_m$. The first inequality of the above is due to Assumption 1 that $\sum_{i=1}^m \lambda_k^i \geq 1$. We next present necessary lemmas.

Lemma 12. *Under the same assumption as Theorem 3, for any gradient descent algorithm updated with composite gradient $\boldsymbol{G}_k\boldsymbol{\lambda}_k$, we have the following inequality*

$$\begin{aligned}
\frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] &\leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})] \\
&\quad + 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).
\end{aligned}$$

Proof. From the L-smoothness of each objective function, we have

$$\lambda_k^i f^i(\boldsymbol{x}_{k+1}) \leq \lambda_k^i\left(f^i(\boldsymbol{x}_k) + \eta_k \nabla f^i(\boldsymbol{x}_k)^\top \boldsymbol{d}_k + \frac{L\eta_k^2}{2}\|\boldsymbol{d}_k\|_2^2\right).$$

Sum up both side for $i = 1, \ldots, m$, and take the expectation on random variable $\boldsymbol{\xi}_k$, we can get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k)] &\leq \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right] + \frac{L\eta_k^2}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \|\boldsymbol{d}_k\|_2^2\right] \\
&\leq \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{d}_k\right] + \frac{LmB\eta_k^2}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right].
\end{aligned}$$

The last inequality is by the Assumption that $\lambda_k^i \leq B$, then $\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i \|\boldsymbol{d}_k\|_2^2\right] \leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m B\|\boldsymbol{d}_k\|_2^2\right] = mB\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right]$. From the result of Lemma 7, we can bound the first term and obtain

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k)] \\
&\leq 2\eta_k mBH\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} - \eta_k \mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] + \frac{LmB\eta_k^2}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{d}_k\|_2^2\right].
\end{aligned}$$

Then, adopting the result from Lemma 8, we know that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k)] &\leq (2\eta_k mBH + 2\eta_k^2 Lm^2B^2H)\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} \\
&\quad + \left(\frac{LmB\eta_k^2}{2} - \eta_k\right)\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).
\end{aligned}$$

By the fact that $\eta_k \leq 1/LmB$, we further have

$$\mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k)] \leq 4mBH\eta_k \sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k] \sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]}$$
$$- \frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

By rearrangement, we therefore have

$$\frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] \leq \mathbb{E}_{\boldsymbol{\xi}_k}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})]$$
$$+ 4mBH\eta_k \sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k] \sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

Take expectation of $\xi_1, \xi_2, \ldots, \xi_n$ on the both sides, we finally get

$$\frac{\eta_k}{2}\mathbb{E}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] \leq \mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})]$$
$$+ 4mBH\eta_k \sqrt{\mathbb{V}[\lambda_k] \sum_{i=1}^m \mathbb{V}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}(\boldsymbol{g}_i).$$

□

Lemma 13. *For CRMOGA algorithm, and under the same assumption with Theorem 3, we have*

$$\sum_{k=1}^n \frac{1}{\eta_k}\mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})] \leq F\sum_{k=2}^n \frac{1}{\eta_k}\mathbb{E}[\sum_{i=1}^m (1-\alpha_k)|\hat{\lambda}_k^i - \lambda_{k-1}^i|] + \frac{2mBF}{\eta_n}.$$

Proof. First, we can decompose the left side as

$$\sum_{k=1}^n \frac{1}{\eta_k}\mathbb{E}[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})]$$
$$= \sum_{k=2}^n \mathbb{E}[\frac{1}{\eta_k}\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \frac{1}{\eta_{k-1}}\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k)] + \mathbb{E}[\frac{1}{\eta_1}\sum_{i=1}^m \lambda_1^i f^i(\boldsymbol{x}_1) - \frac{1}{\eta_n}\sum_{i=1}^m \lambda_n^i f^i(\boldsymbol{x}_{n+1})]$$

By the definition of CRMOGA algorithm, we know that $\lambda_k^i = \alpha_k\lambda_{k-1}^i + (1-\alpha_k)\hat{\lambda}_k^i, i = 1, \ldots, m$, where $\{\hat{\lambda}_k^i\}_{i=1}^m$ are calculated by MOGA algorithms. Then we have the following decomposition for the first term

$$\sum_{k=2}^n \mathbb{E}[\frac{1}{\eta_k}\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \frac{1}{\eta_{k-1}}\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k)]$$
$$= \sum_{k=2}^n \mathbb{E}[\frac{1}{\eta_k}\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \frac{1}{\eta_k}\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k) + \frac{1}{\eta_k}\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k) - \frac{1}{\eta_{k-1}}\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k)]$$
$$= \sum_{k=2}^n \mathbb{E}[\sum_{i=1}^m \frac{1}{\eta_k}(\lambda_k^i - \lambda_{k-1}^i) f^i(\boldsymbol{x}_k)] + \sum_{k=2}^n \mathbb{E}[(\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k)]$$
$$= \sum_{k=2}^n \frac{1}{\eta_k}\mathbb{E}[\sum_{i=1}^m (1-\alpha_k)(\hat{\lambda}_k^i - \lambda_{k-1}^i) f^i(\boldsymbol{x}_k)] + \sum_{k=2}^n \mathbb{E}[(\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\sum_{i=1}^m \lambda_{k-1}^i f^i(\boldsymbol{x}_k)]$$

Combining the above, and by the assumption that $|f^i(\boldsymbol{x}_k)| \leq F, \lambda_k^i, \hat{\lambda}_k^i \in [0, B]$ as well as the nonincreasing for η_k, we therefore have

$$
\begin{aligned}
&\sum_{k=1}^{n} \frac{1}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_{k+1})] \\
&\leq \sum_{k=2}^{n} \frac{1}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} (1-\alpha_k) |\hat{\lambda}_k^i - \lambda_{k-1}^i| |f^i(\boldsymbol{x}_k)|] + \sum_{k=2}^{n} \mathbb{E}[(\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}}) \sum_{i=1}^{m} \lambda_{k-1}^i |f^i(\boldsymbol{x}_k)|] \\
&\quad + \mathbb{E}[\frac{1}{\eta_1} \sum_{i=1}^{m} \lambda_1^i |f^i(\boldsymbol{x}_1)| + \frac{1}{\eta_n} \sum_{i=1}^{m} \lambda_n^i |f^i(\boldsymbol{x}_{n+1})|] \\
&\leq F \sum_{k=2}^{n} \frac{1}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} (1-\alpha_k) |\hat{\lambda}_k^i - \lambda_{k-1}^i|] + mBF \sum_{k=2}^{n} \mathbb{E}[(\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}}) + \frac{mBF}{\eta_1} + \frac{mBF}{\eta_n} \\
&\leq F \sum_{k=2}^{n} \frac{1}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} (1-\alpha_k) |\hat{\lambda}_k^i - \lambda_{k-1}^i|] + \frac{2mBF}{\eta_n}.
\end{aligned}
$$

□

Now, we can present the final proof. From Lemma 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k] \leq m^2 B^2 (1-\alpha_k)^2$. From Assumption 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k)) \leq \sigma^2$. Then plug into Lemma 12, we get

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\sum_{i=1}^{m} \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] \\
&\leq \frac{2}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_{k+1})] + 8m^{5/2} B^2 H \sigma (1-\alpha_k) + L m^3 B^3 \sigma^2 \eta_k.
\end{aligned}
$$

The above is divided with $\eta_k/2$ for the both side. Then, sum up the above inequality from $k=1$ to $k=n$ and take the result from Lemma 13, we have

$$
\begin{aligned}
&\sum_{k=1}^{n} \mathbb{E}\left[\left\|\sum_{i=1}^{m} \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] \\
&\leq \sum_{k=1}^{n} \frac{2}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_{k+1})] + 4m^{5/2} B^2 H \sigma \sum_{k=1}^{n} (1-\alpha_k) + \frac{L m^3 B^3 \sigma^2}{2} \sum_{k=1}^{n} \eta_k \\
&\leq F \sum_{k=2}^{n} \frac{2}{\eta_k} \mathbb{E}[\sum_{i=1}^{m} (1-\alpha_k) |\hat{\lambda}_k^i - \lambda_{k-1}^i|] + \frac{2mBF}{\eta_n} + 4m^{5/2} B^2 H \sigma \sum_{k=1}^{n} (1-\alpha_k) + \frac{L m^3 B^3 \sigma^2}{2} \sum_{k=1}^{n} \eta_k.
\end{aligned}
$$

By averaging the above inequality, we prove the theorem.

E.6 Proof of Corollary 1

Corollary 1. ***(a)*** *Set* $1/mLB \geq \eta_k = O(1/\sqrt{k}), \alpha_k = \max\{1-\eta_k/\eta_1, 1-\eta_k/(\eta_1 k \sum_{i=1}^{m} |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ *in Theorem 2, Algorithm 1 converges with* $O(n^{1/2})$ *rates in the convex case.* ***(b)*** *Set* $1/mLB \geq \eta_k = O(1/\sqrt{k}), \alpha_k = \max\{1-\eta_k/\eta_1, 1-\eta_k/(\eta_1 \sqrt{k} \sum_{i=1}^{m} |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ *in Theorem 3, Algorithm 1* $O(n^{1/4})$ *rates in the non-convex case.*

Proof. For the convex case **(a)**, $\alpha_k = \max\{1-\eta_k/\eta_1, 1-\eta_k/(\eta_1 k\sum_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ means $1-\alpha_k = \min\{\eta_k/\eta_1, \eta_k/(\eta_1 k\sum_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$. Plug in Theorem 2, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\max_{\boldsymbol{x}_k^*\in\mathcal{K}}\min_{\boldsymbol{\lambda}_k^*\in S_m}(\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k)-\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right]\\
&\le\left(\frac{1}{2\eta_n}+\frac{1}{2\eta_1}\right)\frac{D^2}{n}+\frac{DB\sigma m^{3/2}}{n}\sum_{k=1}^{n}(1-\alpha_k)+\frac{6m^{5/2}B^2H\sigma}{n}\sum_{k=1}^{n}(1-\alpha_k)\eta_k\\
&+\frac{m^2B^2(\sigma^2+\sigma H+H^2)}{n}\sum_{k=1}^{n}\eta_k+\frac{2}{n}\sum_{l=1}^{n}l(1-\alpha_{l+1})\sum_{i=1}^{m}|\lambda_l^i-\hat{\lambda}_{l+1}^i|\\
&\le O(\frac{1}{n})+\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})+\sum_{k=1}^{n}O(\frac{1}{k})++\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})++\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})\le O(\sqrt{n}).
\end{aligned}$$

For the non-convex case **(b)**, $\alpha_k = \max\{1-\eta_k/\eta_1, 1-\eta_k/(\eta_1\sqrt{k}\max_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ means $1-\alpha_k = \min\{\eta_k/\eta_1, \eta_k/(\eta_1\sqrt{k}\max_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$. Plug in Theorem 2, we have

$$\begin{aligned}
\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^*\in S_m}\|\nabla\boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right]&\le\frac{F}{n}\sum_{k=2}^{n}\frac{2}{\eta_k}\mathbb{E}[\sum_{i=1}^{m}(1-\alpha_k)|\hat{\lambda}_k^i-\lambda_{k-1}^i|]+\frac{2mBF}{n\eta_n}\\
&+\frac{4m^{5/2}B^2H\sigma}{n}\sum_{k=1}^{n}(1-\alpha_k)+\frac{Lm^3B^3\sigma^2}{2n}\sum_{k=1}^{n}\eta_k\\
&\le\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})+O(\frac{1}{n})+\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})+\sum_{k=1}^{n}O(\frac{1}{\sqrt{k}})\le O(\sqrt{n}).
\end{aligned}$$

□

Remark 1. Also, it is not difficult to verify that setting $\alpha_k = \max\{1-\eta_k/\eta_1, 1-\eta_k/(\eta_1 k\sum_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ also lead to the same convergence in the non-convex case. For simplicity, we present the unified version in the main text, and provide the complete version in the Appendix.

Remark 2. Since multi-objective optimization could reduce to single-objective optimization if the multiple objectives are the same, the convergence rate of stochastic MOO will be no better than single-objective SGD. Note that our convergence rates align with those in single-objective SGD [7, 2], which suggests that our bounds are tight in order. In comparison with stochastic MOGA, we have demonstrated clearly in this paper that stochastic MOGA algorithms fail to converge to the Pareto optimal/critical points in Section 4, so it would be less meaningful to compare them. As for comparing with full-batch MOGA, the comparison is similar with GD and SGD.

E.7 Discussion on Theorem 2 and Theorem 3

Average scheme. The averaging scheme for Theorem 2 and Theorem 3 can be transformed to traditional ones. It just needs to use a typical uniform sampling technique in stochastic optimization [4, 26, 7, 22] to modify the algorithmic output. Specifically, the output is changed from $\boldsymbol{x}_n$ to $\bar{\boldsymbol{x}}_n$ that is uniformly sampled in $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. Then, by this randomness and the linearity of expectation, we can know that

$$\begin{aligned}
\mathbb{E}\left[\max_{\boldsymbol{x}^*\in\mathcal{K}}\min_{\boldsymbol{\lambda}^*\in S_m}(\boldsymbol{\lambda}^{*\top}\boldsymbol{F}(\bar{\boldsymbol{x}}_n)-\boldsymbol{\lambda}^{*\top}\boldsymbol{F}(\boldsymbol{x}^*))\right]&=\mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n}\max_{\boldsymbol{x}_k^*\in\mathcal{K}}\min_{\boldsymbol{\lambda}_k^*\in S_m}(\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k)-\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right]\\
&=\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\max_{\boldsymbol{x}_k^*\in\mathcal{K}}\min_{\boldsymbol{\lambda}_k^*\in S_m}(\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k)-\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right].
\end{aligned}$$

and

$$\mathbb{E}\left[\min_{\boldsymbol{\lambda}^*\in S_m}\|\nabla \boldsymbol{F}(\bar{\boldsymbol{x}}_n)\boldsymbol{\lambda}^*\|^2\right]=\mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n}\min_{\boldsymbol{\lambda}_k^*\in S_m}\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right]=\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^*\in S_m}\|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right].$$

Therefore, we recover the traditional convergence bound for $\bar{\boldsymbol{x}}_n$.

Assumptions of bounded gradient. The assumption of bounded gradients appears very widely in related literature on stochastic optimization. Specifically, in the convex setting, the bounded gradient assumption is used to derive the convergence bound for various famous optimizers, such as Adam (Theorem 4.1 in [24]), Adagrad (Theorem 5 in [8]), AMSGrad (Theorem 4 in [38]), etc. In the non-convex setting, it helps to analyze the stochastic convergence of momentum (Theorem 1 & 2 in [46], Assumption 3 in [4], Assumption 2 in [47], and Assumption 1(ii) in [23]). Additionally, in practice with deep learning, there are many techniques that are designed to avoid gradient explosion, such as weight regularization [37] and gradient clipping [49]. These techniques ensure the boundness of the gradient in practice, which validates this assumption. Furthermore, our theoretical results only require that $\|\nabla f^i(x_k)\|\leq H$ for $i=1,\ldots,m$ and $k=1,\ldots,n$, i.e., the gradients along the optimization path are bounded, which is also supported by our experiments.

Assumptions of bounded function. The assumption of bounded functions has been used in recent studies on non-convex stochastic optimization, such as analyzing SGD (Assumption 2 in [9]), and analyzing SGD+momentum (Assumption in Section 3 of [27]). Actually, this assumption can be further relaxed in our analysis. Specifically, in the convex setting, it can be relaxed to that the function values of Pareto optimal solutions are bounded, i.e., $|f^i(\boldsymbol{x}^*)|\leq F, i=1,\ldots,m$, if $\boldsymbol{x}^*$ is Pareto optimal. The reason is that we only require this condition in Lemma 11. The relaxed assumption is surely satisfied since the optimal solutions naturally have bounded function values. In the non-convex setting, note that in the related papers analyzing SGD (Remark 2 in the analysis of SGD [7]) as well as MOGA (Theorem 3.1 in the analysis of MGDA [11]) with non-convex functions, it is commonly assumed that the initial function values $f^i(\boldsymbol{x}_1), i=1,\ldots,m$ are bounded. Essentially, our analysis only requires that $f^i(\boldsymbol{x}_k), i=1,\ldots,m$ are bounded for $k=1,\ldots,n$, since we only use it in Lemma 13. This assumption is mild in practice as long as the function values of iterates along the practical training path are bounded. In fact, all our experiments indicate that the function values are well located in a bounded regime.

F Convergence Analysis for Strongly Convex Functions

Theorem 6. *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x})$ are all μ-strongly convex and bounded as $f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x})\leq F$, and the distance from sequence to Pareto set is bounded, i.e., $\|\boldsymbol{x}_k-\boldsymbol{x}^*\|\leq D$ for $\boldsymbol{x}^*$ in Pareto set. Set $0\leq\eta_n\leq\ldots\leq\eta_1\leq 1/mLB, \alpha_k\in(0,1]$ for $k=1,\ldots,n$ in Algorithm 1, it achieves*

$$\mathbb{E}\left[\sum_{i=1}^{m}\lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1})-\sum_{i=1}^{m}\lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right]\leq(1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k)-\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+4m^{5/2}B^2H\eta_k(1-\alpha_k)+\frac{Lm^3B^3\sigma^2\eta_k^2}{2}+2F(1-\alpha_{k+1})\sum_{i=1}^{m}|\hat{\lambda}_{k+1}^i-\lambda_k^i|.$$

Proof. By the property of μ-strongly convex ((4.12) from [2]), we have

$$2\mu\left(\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k)-\sum_{i=1}^{m}\lambda_k^i f^i(\boldsymbol{x}_k^*)\right)\leq\left\|\sum_{i=1}^{m}\lambda_k^i\nabla f^i(\boldsymbol{x}_k)\right\|_2^2,$$

where $\boldsymbol{x}_k^* = \arg\min_{\boldsymbol{x}} \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x})$. Recall Lemma 12, we know that

$$\frac{\eta_k}{2}\mathbb{E}_{\boldsymbol{\xi}_k}\left[\left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|_2^2\right] \leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})\right]$$
$$+ 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

By combining the above, we get

$$\eta_k\mu\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] \leq \mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1})\right]$$
$$+ 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

By rearranging the inequality, we further have

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] \leq (1-\eta_k\mu)\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

By the optimality of $\boldsymbol{x}_k^*$, we know that $\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*) \leq \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}^*)$. Therefore, we can get

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta_k\mu)\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i).$$

Further by decomposing, we can obtain

$$\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)$$
$$= \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}^*) - \sum_{i=1}^m (\lambda_k^i - \lambda_{k+1}^i) f^i(\boldsymbol{x}_{k+1}) + \sum_{i=1}^m (\lambda_k^i - \lambda_{k+1}^i) f^i(\boldsymbol{x}_{k+1}^*).$$

By the definition of Algorithm 1 and the boundness of gradient function. We know that $\lambda_{k+1}^i - \lambda_k^i = (1-\alpha_{k+1})(\hat{\lambda}_{k+1}^i - \lambda_k^i)$ and $|f^i(\boldsymbol{x}_k)| \leq F, |f^i(\boldsymbol{x}_k^*)| \leq F$. Then, we can easily get that $-(\lambda_k^i - \lambda_{k+1}^i) f^i(\boldsymbol{x}_{k+1}) \leq F(1-\alpha_{k+1})|\hat{\lambda}_{k+1}^i - \lambda_k^i|$ and $(\lambda_k^i - \lambda_{k+1}^i) f^i(\boldsymbol{x}_{k+1}^*) \leq F(1-\alpha_{k+1})|\hat{\lambda}_{k+1}^i - \lambda_k^i|$. Thus, we have

$$\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)$$
$$\leq \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_{k+1}^*) + 2F\sum_{i=1}^m (1-\alpha_{k+1})|\hat{\lambda}_{k+1}^i - \lambda_k^i|.$$

Then combining the above, we now obtain

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta_k\mu)\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4mBH\eta_k\sqrt{\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k]\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{g}^i(\boldsymbol{x}_k)]} + \frac{Lm^2B^3\eta_k^2}{2}\sum_{i=1}^m \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}_i) + 2F\sum_{i=1}^m (1-\alpha_{k+1})|\hat{\lambda}_{k+1}^i - \lambda_k^i|.$$

From Lemma 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}[\lambda_k] \leq m^2B^2(1-\alpha_k)^2$. From Assumption 2, we know that $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{g}^i(\boldsymbol{x}_k)) \leq m\sigma^2$. Then plug into the above, we get

$$\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta_k\mu)\mathbb{E}_{\boldsymbol{\xi}_k}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4m^{5/2}B^2H\eta_k(1-\alpha_k) + \frac{Lm^3B^3\sigma^2\eta_k^2}{2} + 2F(1-\alpha_{k+1})\sum_{i=1}^m |\hat{\lambda}_{k+1}^i - \lambda_k^i|.$$

Taking total expectations, this yields

$$\mathbb{E}\left[\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4m^{5/2}B^2H\eta_k(1-\alpha_k) + \frac{Lm^3B^3\sigma^2\eta_k^2}{2} + 2F(1-\alpha_{k+1})\sum_{i=1}^m |\hat{\lambda}_{k+1}^i - \lambda_k^i|.$$

□

We now are able to provide the convergence analysis for Algorithm 1 with fixed stepsizes.

Corollary 2. *Set* $0 < \eta_k = \eta \leq \min\{\frac{1}{mLB}, \frac{1}{L}\}$ *and* $\alpha_k = \max\{1-\eta, 1-\eta^2/(\sum_{i=1}^m |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ *in Theorem 6. Algorithm 1 linearly converges to a solution near the Pareto optimal, and the expected gap limits to* $\eta M/\mu$*, where constant* $M = 4m^{5/2}B^2H + \frac{Lm^3B^2\sigma^2}{2} + 2F$.

Proof. From the parameter setting, we know that $1-\alpha_k = \min\{\eta, \eta^2/(\sum_{i=1}^m |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$, which suggests that $1-\alpha_k \leq \eta$ and $(1-\alpha_{k+1})\sum_{i=1}^m |\hat{\lambda}_{k+1}^i - \lambda_k^i| \leq \eta^2$. By Theorem 6, we have

$$\mathbb{E}\left[\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4m^{5/2}B^2H\eta(1-\alpha_k) + \frac{Lm^3B^3\sigma^2\eta^2}{2} + 2F(1-\alpha_{k+1})\sum_{i=1}^m |\hat{\lambda}_{k+1}^i - \lambda_k^i|$$
$$\leq (1-\eta\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta^2(4m^{5/2}B^2H + \frac{Lm^3B^2\sigma^2}{2} + 2F)$$
$$= (1-\eta\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta^2 M,$$

where $M = 4m^{5/2}B^2H + \frac{Lm^3B^2\sigma^2}{2} + 2F$. Then, subtracting the constant $\eta M/\mu$ from both sides, we obtain

$$\mathbb{E}\left[\sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^m \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] - \frac{\eta M}{\mu}$$
$$\leq (1-\eta\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta^2 M - \frac{\eta M}{\mu}$$
$$= (1-\eta\mu)\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] - \frac{\eta M}{\mu}(1-\eta\mu)$$
$$= (1-\eta\mu)\left(\mathbb{E}\left[\sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^m \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] - \frac{\eta M}{\mu}\right).$$

Observe that the above is a contraction inequality, by the fact that

$$0 < \eta\mu \leq \frac{\mu}{L} \leq 1.$$

The second inequality is by the range of η. The result thus follows by applying the above contraction inequality repeatedly through iteration $k = 1, \ldots, n$.

$$\mathbb{E}\left[\sum_{i=1}^{m} \lambda_n^i f^i(\boldsymbol{x}_n) - \sum_{i=1}^{m} \lambda_n^i f^i(\boldsymbol{x}_n^*)\right]$$
$$\leq (1-\eta\mu)^{k-1}\left(\mathbb{E}\left[\sum_{i=1}^{m} \lambda_1^i f^i(\boldsymbol{x}_1) - \sum_{i=1}^{m} \lambda_1^i f^i(\boldsymbol{x}_1^*)\right] - \frac{\eta M}{\mu}\right) + \frac{\eta M}{\mu}.$$

If $n \to \infty$, then the suboptimal gap for the solution will limit to $\frac{\eta M}{\mu}$. □

Remark. Here, the algorithm does not converge to Pareto optimal but is a suboptimal solution near an optimal solution. This result aligns with the one in single-objective stochastic optimization, and thus is tight in order. Note that this gap is unavoidable, as is the case in single-objective SGD, because the stepsize needs to tradeoff the stochastic variance and the convergence rate (the discussion of Theorem 4.6 in [2]). Additionally, the suboptimal gap is related to the scale of the stepsize η, and hence we can set sufficiently small η to get better convergence.

To guarantee the convergence to Pareto optimal solutions, we can use a diminishing stepsize as the case in single-objective stochastic optimization [2]. The result is as follows.

Corollary 3. *Set $\eta_k = \frac{\beta}{k+\gamma}, \beta > \frac{1}{\mu}$ where $\gamma > 0$ such that $\eta_k \leq \frac{1}{mLB}$, and $\alpha_k = \max\{1 - \eta_k/\eta_1, 1 - \eta_k^2/(\eta_1 \sum_{i=1}^m |\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ in Theorem 6. Algorithm 1 linearly converges to Pareto optimal with $O(1/n)$ rates.*

Proof. Specifically, we prove the following result for $k = 1, \ldots, n$

$$\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] \leq \frac{v}{k+\gamma},$$

where v satisfies that $v = \max\{\frac{\beta^2 M}{\beta\mu - 1}, (1+\gamma)(\sum_{i=1}^m \lambda_1^i f^i(\boldsymbol{x}_1) - \sum_{i=1}^m \lambda_1^i f^i(\boldsymbol{x}_1^*))\}$. From Theorem 6, we know that

$$\mathbb{E}\left[\sum_{i=1}^{m} \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}) - \sum_{i=1}^{m} \lambda_{k+1}^i f^i(\boldsymbol{x}_{k+1}^*)\right] \leq (1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right]$$
$$+ 4m^{5/2}B^2 H\eta_k(1-\alpha_k) + \frac{Lm^3B^3\sigma^2\eta_k^2}{2} + 2F(1-\alpha_{k+1})\sum_{i=1}^{m}|\hat{\lambda}_{k+1}^i - \lambda_k^i|$$
$$\leq (1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta_k^2\left(\frac{4m^{5/2}B^2H}{\eta_1} + \frac{Lm^3B^2\sigma^2}{2} + \frac{2F}{\eta_1}\right) + \frac{2F\eta_{k+1}^2}{\eta_1}$$
$$\leq (1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta_k^2\left(\frac{4m^{5/2}B^2H}{\eta_1} + \frac{Lm^3B^2\sigma^2}{2} + \frac{4F}{\eta_1}\right)$$
$$= (1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] + \eta_k^2 M,$$

where the constant in the above is defined as $M = \frac{4m^{5/2}B^2H}{\eta_1} + \frac{Lm^3B^2\sigma^2}{2} + \frac{4F}{\eta_1}$ for simplicity. The last inequality is by the fact that $\eta_k \geq \eta_{k+1}$. We then prove the corollary by mathematical induction. First, the parameter setting here ensures the result holds for $k = 1$. Next, we assume the inequality of the theorem holds for k, which holds

$$\mathbb{E}\left[\sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k) - \sum_{i=1}^{m} \lambda_k^i f^i(\boldsymbol{x}_k^*)\right] \leq \frac{v}{k+\gamma}.$$

Then by the parameter setting, for the case of $k+1$, we have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{m}\lambda_{k+1}^{i}f^{i}(\boldsymbol{x}_{k+1})-\sum_{i=1}^{m}\lambda_{k+1}^{i}f^{i}(\boldsymbol{x}_{k+\gamma}^{*})\right]\\
&=(1-\eta_k\mu)\mathbb{E}\left[\sum_{i=1}^{m}\lambda_{k}^{i}f^{i}(\boldsymbol{x}_{k})-\sum_{i=1}^{m}\lambda_{k}^{i}f^{i}(\boldsymbol{x}_{k}^{*})\right]+\eta_k^2M\\
&\leq(1-\frac{\beta\mu}{k+\gamma})\frac{v}{k+\gamma}+\frac{\beta^2M}{(k+\gamma)^2}\\
&=\frac{k+\gamma-\beta\mu}{(k+\gamma)^2}v+\frac{\beta^2M}{(k+\gamma)^2}\\
&=\frac{k+\gamma-1}{(k+\gamma)^2}v+\frac{1-\beta\mu}{(k+\gamma)^2}v+\frac{\beta^2M}{(k+\gamma)^2}\\
&=\frac{k+\gamma-1}{(k+\gamma)^2}v-\frac{1}{(k+\gamma)^2}((\beta\mu-1)v-\beta^2M)\\
&\leq\frac{k+\gamma-1}{(k+\gamma)^2}v,
\end{aligned}
$$

where the last inequality is by the condition of v. By the fact that $(a-1)(a+1)\leq a^2$ for $a>0$, we have $\frac{k+\gamma-1}{(k+\gamma)^2}\leq\frac{k+\gamma-1}{(k+\gamma-1)(k+1+\gamma)}=\frac{1}{k+1+\gamma}$. Plug in the above, we can prove the result for $k+1$. Thus, we theorem holds for $k=1,\ldots,n$, which suggests an $O(1/n)$ convergence. □

Remark 1. The result from Corollary 3 shows the convergence rate is aligned with single-objective SGD with adaptive stepsize, suggesting that Algorithm 1 is tight in order with strongly convex functions.

Remark 2. The bounded function assumption in Theorem 6 can be relaxed to the combination of the assumptions in the convex and non-convex settings. Specifically, the function values of Pareto optimal solutions are bounded, i.e., $|f^i(\boldsymbol{x}^*)|\leq F, i=1,\ldots,m$, and the function values of iterates along the practical training path are bounded, i.e., $f^1(\boldsymbol{x}_k),\ldots,f^m(\boldsymbol{x}_k)\leq F$ for $k=1,\ldots,n$.

G Additional Experiments and More Details

G.1 More Insights of CR-MOGA

How the non-convergence reflects in practice. In fact, the similar situation reflected by the counter-example is not rare in practice, especially when the stochastic noises are large. For example, we have added an additional experiment of MultiMNIST by setting batch size from 8 to 2, representing increasing levels of stochastic noises (a smaller batch size introduces a larger stochastic noise in the mini-batch gradient). We report the training losses for the final outputs of MGDA and CR-MGDA as follows, representing the comparison of MOGA and CR-MOGA. From results of Table 2, we observe that the loss of vanilla MGDA becomes significantly larger when the batch size is smaller, while CR-MGDA appears not sensitive to the batch size. This possible reason is that the convergence issue of MOGA becomes more severe with larger stochastic noise, which severely degrades the performance of MOGA. Since CR-MOGA is proven to converge to the Pareto optimal/critical, it is much more robust to stochastic noises. Thus, in this regard, our method is more practical for real-world applications.

Comparison with the full-batch MOGA. Although there are no guarantees for the closeness between $\boldsymbol{\lambda}$ calculated by CR-MOGA and the actual $\boldsymbol{\lambda}$ calculated by MOGA with full-batch gradients. As shown in Figure 4, we empirically observe that the $\boldsymbol{\lambda}_k$ of CR-MOGA is very close to the actual $\boldsymbol{\lambda}_k^*$ yielded by the MOGA with full batch gradients. Specifically, the gaps between $\boldsymbol{\lambda}_k$ of CR-MOGA and actual $\boldsymbol{\lambda}_k^*$ are below 0.01 with low error bars in the toy example, while the gaps of vanilla MOGA algorithms are all averagely larger than 0.10 with large error bars.

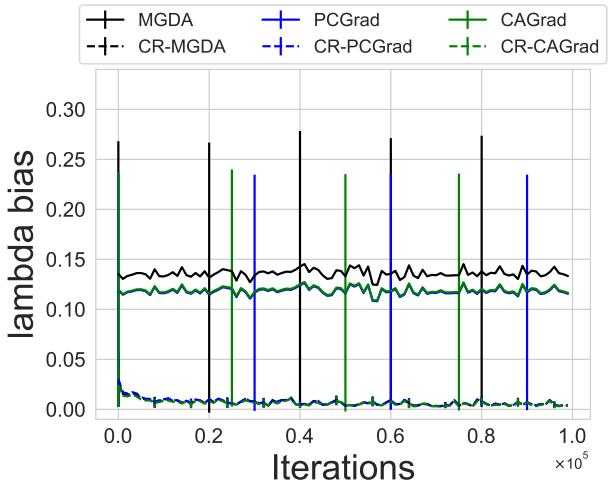


Figure 4: Experiments on the illustrative example in Section 4. We measure the bias between $\boldsymbol{\lambda}$ calculated by CR-MOGA and the actual $\boldsymbol{\lambda}$ calculated by MOGA with full-batch gradients. We report the mean and std.

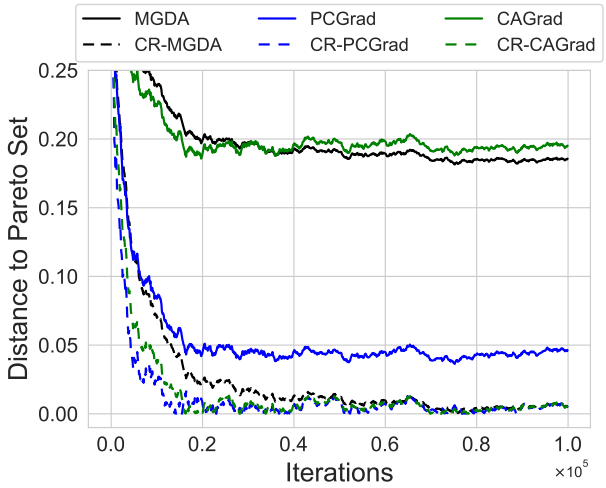


Figure 5: Simulation with Gaussian noise.

G.2 Additional Stochastic Convex Simulation

In this simulation, we establish a similar setting with Section 4, where we aim to optimize $\boldsymbol{x} \in \mathbb{R}^2$ to attain smaller distances to two fixed points $\boldsymbol{a} = [-3, 0], \boldsymbol{b} = [3, 0]$. The first objective is $f^1(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{a}\|_2^2$, and the second one is $f^2(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{b}\|_2^2$. The stochastic gradient for each objective can be represented as $g^1(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{a} + \boldsymbol{\xi}^1$ and $g^2(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{b} + \boldsymbol{\xi}^2$. The stochastic noises $\boldsymbol{\xi}^1, \boldsymbol{\xi}^2$ are two dimensional random noise with zero-mean Gaussian distribution, where the covariance matrix for $\boldsymbol{\xi}^1$ is $[[3, 2], [2, 3]]$ and for $\boldsymbol{\xi}^2$ is $[[3, -2], [-2, 3]]$.

Figure 5 demonstrates the distance to Pareto set. We observe that all three MOGA algorithms can not converge to the Pareto optimal points, while all CRMOGA algorithms can approach the Pareto set. This result is aligned with the result in Section 4, and the more general Gaussian noise shows that the non-convergence is not special in stochastic multiobjective optimization.

G.3 Implementation Details of Experiments

The implementation for CRMOGA is very simple, just by adding one additional line that exponentially smoothes the composite weights to MOGA algorithms. Note that this mechanism does not require additional computational cost and is flexible to be implemented in MOGA algorithms.