A Multi-objective / Multi-task Learning Framework Induced by Pareto Stationarity

Michinari Momma [1] **Chaosheng Dong** [1] **Jia Liu** [2]

Abstract

Multi-objective optimization (MOO) and multi-task learning (MTL) have gained much popularity with prevalent use cases such as production model development of regression / classification / ranking models with MOO, and training deep learning models with MTL. Despite the long history of research in MOO, its application to machine learning requires development of solution strategy, and algorithms have recently been developed to solve specific problems such as discovery of any Pareto optimal (PO) solution, and that with a particular form of preference. In this paper, we develop a novel and generic framework to discover a PO solution with multiple forms of preferences. It allows us to formulate a generic MOO / MTL problem to express a preference, which is solved to achieve both alignment with the preference and PO, at the same time. Specifically, we apply the framework to solve the weighted Chebyshev problem and an extension of that. The former is known as a method to discover the Pareto front, the latter helps to find a model that outperforms an existing model with only one run. Experimental results demonstrate not only the method achieves competitive performance with existing methods, but also it allows us to achieve the performance from different forms of preferences.

1. Introduction

Multi-objective optimization (MOO) (Kaisa, 1999; Zhang & Li, 2007) and multi-task learning (MTL) (Caruana, 1997) have gained much popularity in machine learning (ML) applications. For example, MOO has been used in production system in search ranking (Momma et al., 2019; 2020; Carmel et al., 2020) and recommendation (Lin et al., 2019a). MTL is one of the common methods to train a deep learning model (Ruder, 2017). Since the two areas share some similarity in its core: trade-off among tasks / objectives, key concepts such as Pareto optimality (PO) and methodologies developed in MOO can also be applicable to MTL. One prominent example is multiple gradient descent algorithm (MGDA) (Désidéri, 2012). MGDA casts the Pareto stationarity (PS) problem that is a necessary condition of PO as a norm minimization problem of convex combination of gradient vectors over all objectives. The resulting gradient is the steepest-descent direction toward PS. This framework is well suited for ML problems where functional optimization via gradient methods is prevalent. The first adaptation of MGDA to MTL (Sener & Koltun, 2018) demonstrates it is able to generate MTL model that outperforms single task baselines. However, although expected by design, the MGDA based solution is found not being able to generate diversified models, or models that meet user preference (e.g., ratio between objective values). Several works address the issue by imposing additional constraints into the MGDA optimization problem (Lin et al., 2019b; Ma et al., 2020; Lin et al., 2019a).

In this paper, we propose a different approach to incorporate preferences. Instead of starting from MGDA and modify it, we start from formulating an optimization problem to incorporate preference, then derive MGDA-like component in the problem. The rationale of this approach is based on the fact that what MGDA is solving for is only a portion of some optimization problem. This is analogous to the more general Karush-Kuhn-Tucker (KKT) conditions where the stationarity condition is only a partial component of the full optimality conditions, which consists of primal / dual feasibility and complementarity in addition to the stationarity. We apply this analogy to the PS problems in this paper.

As applications in our framework, we formulate the weighted Chebyshev (WC) problem (Kaisa, 1999). This problem is popular in the traditional MOO literature and can be solved to discover the Pareto front (PF), a set of PO points, by minimizing the ℓ_∞ norm between the *ideal point* – infimum of each objective, and a PO point on PF. The weight in WC represents preference in terms of ratio

[1]Amazon.com Inc. [2]The Ohio State University. Correspondence to: Michinari Momma <michi@amazon.com>, Chaosheng Dong <chaosd@amazon.com>, Jia Liu <liu@ece.osu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

between objectives and it tries to find a PO solution aligned with the preference. WC is known to be able to discover the entire points in PF. This problem, however, is not suited for ML applications as is – WC defines only the optimization problem, and essential components in ML modeling such as generation of gradient, and optimization strategy to handle complex loss functions, etc., need to be specified. We address the issue by incorporating the PS condition into the original optimization problem in WC, and develop an optimization strategy to solve the problem efficiently.

In many ML applications especially in industrial settings, we often need to retrain a model with a fresh dataset. The retrained model should achieve at least the same (if not better) performance over all objectives / tasks as the existing model trained with older dataset. Further, from such models, we want to select a model that has desired trade-offs over the objectives / tasks, to replace the existing model. This requires exploration over a specific portion of PF. Existing methods such as Momma et al. (2020) is not designed to optimize such a problem, and requires $\Omega(m)$ explorations at least, where m is the number of objectives / tasks. We address the challenge by extending the WC method to allowing an arbitrary reference point / model (e.g., a baseline model) in place of the ideal point in the formulation. Hence, *by design*, the extended WC allows us to explore the specific PF pivoted on the reference model. Application of the extended WC method would help modelers to be free from the labor intensive tuning over multi-objectives / multi-tasks, which typically takes at least a few days – reduction from $\Omega(m)$ to $O(1)$. This is one of the biggest motivations for us to develop the extended WC method.

The paper is organized as follows; Section 2 discusses the related work. Section 3 is devoted to developing the proposed method, Section 4 shows experimental results, and Section 5 concludes this paper.

2. Related work

As a MOO method, MGDA (Désidéri, 2012; Fliege & Svaiter, 2000) is found to be well suited to MTL applications as it generates a gradient vector to discover PO solutions, which is directly used in the existing gradient descent framework. Sener & Koltun (2018) first leveraged MGDA for MTL problems. However, MGDA alone does not suffice since it may not generate models that are aligned with a preference, not to mention diverse solution to cover the full PF. To address the issue, Lin et al. (2019b) proposed a method to diversify the MGDA solution by imposing constraints to split the loss space. Further, Lin et al. (2019a) imposed constraints directly on the Lagrange multiplier to influence the MGDA solution. These methods are built based on the MGDA problem with additional constraints, and do not address the fundamental point of MGDA being a method only based on PS. In contrast, our method formulates a full optimization problem to incorporate preference and develop an algorithm with the optimality conditions. In this regard, the Exact Pareto Optimal (EPO) Search (Mahapatra & Rajan, 2020) is closer to our work. EPO directly formulates the optimization problem to optimize 1) uniformity, which is defined as the KL divergence between the weighted loss function and unity, and 2) PO. The optimization goal 1) is similar to the WC problem. However, the algorithm is designed specific to solving the problem with requirements such as positivity on less, etc., and it is not straightforward to extend it into more general problems such as handling arbitrary reference points.

Another research direction is to generate the full PF. There are several approaches such as perturbing the PO / MGDA solution (Ma et al., 2020; Liu et al., 2021), building a hypernetwork (Lin et al., 2020; Navon et al., 2021) or single nextwork (Ruchte & Grabocka, 2021) to learn the entire PF. Unlike these methods, our focus is to find PO solutions that are aligned with user given preferences.

Aside from MGDA-like methods, Momma et al. (2019; 2020) proposed an ϵ-constraint (EC) problem, which is also popular in traditional MOO literature, for search ranking. They apply the augmented Lagrangian (AL) method to solve the hard constrained EC problem. Gong et al. (2021) proposed a dynamic barrier gradient descent (DBGD) approach for EC and lexicographic optimization problems. These methods have parameters to control smoothness of the optimization steps, which has to be tuned to ensure best performance. In contrast, our method auto-tunes such a parameter.

Note, similar to related works in gradient based methods, such as Liu et al. (2021), etc., we do not compare with derivative-free MOO / MTL methods (e.g., those based on evolutionary algorithms or Bayesian optimization) which could fail to solve large-scale MOO / MTL problems because of the lack of gradient information.

3. Methodology

In this section, we introduce the basic settings of the investigated problem and the development of our proposed method and its extension. We summarize the main notations used throughout this paper in Table 1.

3.1. An MOO / MTL problem

Suppose we have a MOO / MTL problem where we minimize a vector of loss of m objectives / tasks: $\boldsymbol{l}(\boldsymbol{x}) = [l_1(\boldsymbol{x}), \dots, l_m(\boldsymbol{x})]^\intercal \in \mathbb{R}^m$ over the vector of model parameters $\boldsymbol{x} \in \mathbb{R}^n$ with $n \gg m$ in ML:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \boldsymbol{l}(\boldsymbol{x}). \tag{1}$$

Table 1. List of notation

Notation	Definition
m	Number of objectives/tasks considered in (1)
n	Deimension of model parameters considered in (1)
$\boldsymbol{e} \in \mathbb{R}^m$	A vector of ones
$\boldsymbol{x} \in \mathbb{R}^n$	Model parameters of the MOO/MTL problem
$\boldsymbol{l}(\boldsymbol{x}) \in \mathbb{R}^m$	Vector of losses: $\boldsymbol{l}(\boldsymbol{x}) = [l_1(\boldsymbol{x}), \ldots, l_m(\boldsymbol{x})]^\intercal$
$\boldsymbol{\alpha} \in \mathbb{R}^m_+$	Coefficients for combining the gradients where each of the components is non-negative
$\boldsymbol{r} \in \mathbb{R}^m_{++}$	Preference vector of the objectives/tasks
$\mathrm{diag}(\sqrt{\boldsymbol{r}})$	Diagonal matrix whose diagonal element is given by $\sqrt{\boldsymbol{r}} = [\sqrt{r_i}, ..., \sqrt{r_m}]^\intercal$
$\boldsymbol{G}(\boldsymbol{x}) \in \mathbb{R}^{n \times m}$	$\boldsymbol{G}(\boldsymbol{x}) = \nabla \boldsymbol{l}(\boldsymbol{x}) = [\nabla l_1(\boldsymbol{x}), \ldots, \nabla l_m(\boldsymbol{x})]$
$\boldsymbol{K} \in \mathbb{R}^{m \times m}$	$\boldsymbol{K} = \sqrt{\boldsymbol{G}^\intercal \boldsymbol{G}}$
$\boldsymbol{K}_r \in \mathbb{R}^{m \times m}$	$\boldsymbol{K}_r = \mathrm{diag}(\sqrt{\boldsymbol{r}}) \boldsymbol{K} \mathrm{diag}(\sqrt{\boldsymbol{r}})$
$u \in \mathbb{R}_+$	Non-negative trade-off constant in (7) and (13)
$\boldsymbol{d} \in \mathbb{R}^m$	Dual variable of (7)
$\mathcal{C}$	Set of second order cones; e.g., $(\boldsymbol{K}_r \boldsymbol{\alpha}, \gamma) \in \mathcal{C}$ implies $\|\boldsymbol{K}_r \boldsymbol{\alpha}\|_2 \leq \gamma$.
$\tilde{\boldsymbol{b}} \in \mathbb{R}^m$	Loss function value of the reference model
$\boldsymbol{v} \in \mathbb{R}^m_{++}$	A constant vector where each of the components is positive
$\boldsymbol{w} \in \mathbb{R}^m_{++}$	$\boldsymbol{w} = \boldsymbol{v}/(1 + \boldsymbol{e}^\intercal \boldsymbol{v})$ where each of the components is positive
$\boldsymbol{z} \in \mathbb{R}^m$	Dual variable of XWC-MGDA
p	The optimal objective value of (13)

When objectives compete with each other, there is a trade-off between the objectives. As a result, there does not exist an "optimal model" that dominates all other models in every single objective. Below, we review some key concepts that characterizes optimality over multi-objectives / tasks.

Pareto dominance: A solution $\hat{\boldsymbol{x}}$ dominates another solution $\boldsymbol{x}$ if and only if $l_i(\hat{\boldsymbol{x}}) \leq l_i(\boldsymbol{x}), \forall i \in I$ and $l_j(\hat{\boldsymbol{x}}) < l_j(\boldsymbol{x}), \exists j \in I$, where $I = \{1, \ldots, m\}$ denotes the index set of all objectives / tasks.

Pareto optimality: A solution $\hat{\boldsymbol{x}}$ is a Pareto optimal (PO) point if there is no other solution that dominates $\hat{\boldsymbol{x}}$. The set of all Pareto optimal points is said to be the Pareto optimal set and the corresponding set of loss function values is said to be the Pareto front.

Weak Pareto optimality: A solution $\hat{\boldsymbol{x}}$ is said to be a weak Pareto optimal point if there is no other solution $\boldsymbol{x}$ such that $l_i(\boldsymbol{x}) < l_i(\hat{\boldsymbol{x}}), \forall i \in I$. While PO mandates the existence of strict inequality relationship, weak PO allows equality.

3.2. MGDA

MGDA aims to solve the **Pareto stationarity** condition – a convex combination of the gradient vectors is zero, i.e., $\boldsymbol{G}(\boldsymbol{x})\boldsymbol{\alpha} = \boldsymbol{0},\ \boldsymbol{\alpha} \geq \boldsymbol{0},\ \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1$, where $\boldsymbol{e}$ is a vector of ones, and $\boldsymbol{G}(\boldsymbol{x}) \equiv \nabla \boldsymbol{l}(\boldsymbol{x}) = [\nabla l_1(\boldsymbol{x}), \ldots, \nabla l_m(\boldsymbol{x})] \in \mathbb{R}^{n \times m}$, by minimizing the ℓ_2 norm $\|\boldsymbol{G}(\boldsymbol{x})\boldsymbol{\alpha}\|_2$. Since $n \gg m$, it is not efficient to directly use $\boldsymbol{G}$ to compute $\|\boldsymbol{G}(\boldsymbol{x})\boldsymbol{\alpha}\|_2$. Instead, we use the matrix square root of $\boldsymbol{G}$: $\boldsymbol{K} \equiv \sqrt{\boldsymbol{G}^\intercal \boldsymbol{G}} \in \mathbb{R}^{m \times m}$, so $\|\boldsymbol{G}(\boldsymbol{x})\boldsymbol{\alpha}\|_2 = \sqrt{\boldsymbol{\alpha}^\intercal \boldsymbol{G}^\intercal \boldsymbol{G} \boldsymbol{\alpha}} = \|\boldsymbol{K}\boldsymbol{\alpha}\|_2$. Using $\boldsymbol{K}$, the Pareto stationarity condition that we work with throughout this paper is written by

$$\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{0},\ \boldsymbol{\alpha} \geq \boldsymbol{0},\ \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1. \tag{2}$$

Finally, MGDA solves the following problem (Désidéri, 2012):

$$\min_{\gamma, \boldsymbol{\alpha} \in \mathbb{R}^m_+} \gamma \ \ s.t.\ \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1,\ \|\boldsymbol{K}\boldsymbol{\alpha}\|_2 \leq \gamma. \tag{3}$$

The primal problem of MGDA (Fliege & Svaiter, 2000) by taking the dual of (3) is given as follows[1]:

$$\min_{\rho, \boldsymbol{d} \in \mathbb{R}^m} \rho \ \ s.t.\ \boldsymbol{K}\boldsymbol{d} \leq \rho \boldsymbol{e},\ \|\boldsymbol{d}\|_2 \leq 1, \tag{4}$$

which is interpreted as the minimax problem of projection of gradient vectors.

3.3. Weighted Chebyshev – MGDA

3.3.1. WC-MGDA FORMULATION

Suppose we have the WC problem, which is to find a PO point with a preference vector $\boldsymbol{r} \in \mathbb{R}^m_{++}$ by minimizing the ℓ_∞-norm of weighted loss:

$$\min_{\rho, \boldsymbol{x} \in \mathbb{R}^n} \rho \quad \text{s.t. } \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) \leq \rho \boldsymbol{e}, \tag{5}$$

where we use the origin as the ideal point, WLOG. By KKT stationarity condition on ρ and $\boldsymbol{x}$, and $\boldsymbol{\alpha}$ being a Lagrange multiplier on the constraints, we have the Wolfe dual:

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m_+, \boldsymbol{x} \in \mathbb{R}^n} \quad & \boldsymbol{\alpha}^\intercal (\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})) \\ s.t. \quad & \boldsymbol{K}_r \boldsymbol{\alpha} = \boldsymbol{0},\ \boldsymbol{\alpha} \geq 0,\ \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1, \end{aligned} \tag{6}$$

[1]Note this is a modified version of the problem in Fliege & Svaiter (2000), derived by taking the dual of (3).

where we incorporate $\boldsymbol{r}$ into $\boldsymbol{K}_r \equiv \text{diag}(\sqrt{\boldsymbol{r}})\boldsymbol{K}\text{diag}(\sqrt{\boldsymbol{r}})$. The following lemma ensures Pareto stationarity in (6);

Lemma 3.1. *Constraints in* (6) *implies the Pareto stationarity condition* (2)*.*

Proof. Consider $\boldsymbol{K}_r\boldsymbol{\alpha} = \text{diag}(\sqrt{\boldsymbol{r}})\boldsymbol{K}\text{diag}(\sqrt{\boldsymbol{r}})\boldsymbol{\alpha} = \boldsymbol{0}$. Since $\boldsymbol{r} \in \mathbb{R}^m_{++}$, we can divide it by $\text{diag}(\sqrt{\boldsymbol{r}^{-1}})$ from left, yielding $\boldsymbol{K}\text{diag}(\sqrt{\boldsymbol{r}})\boldsymbol{\alpha} = \boldsymbol{0}$. By defining $\text{diag}(\sqrt{\boldsymbol{r}})\boldsymbol{\alpha} = \tilde{\boldsymbol{\alpha}}$, we get $\boldsymbol{K}\tilde{\boldsymbol{\alpha}} = 0,\ \tilde{\boldsymbol{\alpha}} \geq 0$.

Further, $\boldsymbol{\alpha} \geq 0$ and $\boldsymbol{e}^\intercal\boldsymbol{\alpha} = 1$ implies $\exists i$, s.t., $\alpha_i > 0$ (this can be proved by contradiction assuming $\boldsymbol{\alpha} = \boldsymbol{0}$), which also implies $\exists i$ s.t., $\tilde{\alpha}_i > 0$. Therefore, by rescaling $\tilde{\tilde{\boldsymbol{\alpha}}} = \tilde{\boldsymbol{\alpha}}/\boldsymbol{e}^\intercal\tilde{\boldsymbol{\alpha}}$, we get the Pareto stationary condition: $\boldsymbol{K}\tilde{\tilde{\boldsymbol{\alpha}}} = 0,\ \boldsymbol{e}^\intercal\tilde{\tilde{\boldsymbol{\alpha}}} = 1,\ \tilde{\tilde{\boldsymbol{\alpha}}} \geq 0$. □

As $\boldsymbol{K}_r\boldsymbol{\alpha} = \boldsymbol{0}$ may not be met at sub-optimality (i.e., during optimization algorithm), we try to minimize the norm of $\boldsymbol{K}_r\boldsymbol{\alpha}$. Using ℓ_2 norm on $\boldsymbol{K}_r\boldsymbol{\alpha}$ and a trade-off constant $u > 0$, we have a working version of Wolfe dual:

$$\begin{aligned} \max_{\gamma,\boldsymbol{\alpha}\in\mathbb{R}^m_+,\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{\alpha}^\intercal(\boldsymbol{r}\odot\boldsymbol{l}(\boldsymbol{x})) - u\gamma \\ s.t. \quad & \boldsymbol{e}^\intercal\boldsymbol{\alpha} = 1,\ \|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq \gamma. \end{aligned} \tag{7}$$

Pleasantly, (7) can be seen as an extension of MGDA where the objective function is specified to find a PS point along the preference vector $\boldsymbol{r}$ (i.e., incorporating a WC objective in MGDA). The use of norm instead of squares of norm ensures same scaling of both terms in the objective function. Problem (7) is referred to as the dual of weighted Chebyshev - MGDA (WC-MGDA) hereafter.

When $\boldsymbol{x}$ is fixed, (7) can be seen as a second-order cone program (SOCP) (Boyd & Vandenberghe, 2004; Alizadeh & Goldfarb, 2001) with $\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq \gamma$ being the second-order constraint (SOC): $(\boldsymbol{K}_r\boldsymbol{\alpha}, \gamma) \in \mathcal{C}$ where $\mathcal{C}$ denotes the set of SOC. By using SOCP duality, we further modify the problem by deriving the dual of (7), i.e., *dual of dual*:

$$\min_{\rho,\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{d}\in\mathbb{R}^m} \rho \ \ s.t.\ \boldsymbol{r}\odot\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{K}_r\boldsymbol{d} \leq \rho\boldsymbol{e},\ \|\boldsymbol{d}\|_2 \leq u. \tag{8}$$

This is a primal of WC-MGDA. It is now clear that this problem can be seen as a mix of WC problem (5) and primal of MGDA (4): it tries to optimize both WC and PO by minimizing the maximum element of $\boldsymbol{r}\odot\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{K}_r\boldsymbol{d}$.

Since SOCP is convex, it is known to enjoy strong duality under the KKT conditions: 1) primal-dual feasibility and 2) complementary slackness holds. In this case, the complementarity is given by $\boldsymbol{\alpha}^\intercal\boldsymbol{K}_r\boldsymbol{d} = \gamma u$. By the SOCP duality, we can get primal variables such as ρ and $\boldsymbol{d}$ once we solve the dual $\boldsymbol{\alpha}$ in (7), and vice versa on the dual variables from (8). However, when $\boldsymbol{x}$ is not fixed, we note that (7) and (8) are non-convex. Similar to other gradient based methods (Sener & Koltun, 2018; Lin et al., 2019b; Mahapatra & Rajan, 2020), we separate the optimization w.r.t. $\boldsymbol{x}$ and other variables. Namely, we conduct the gradient descent on $\boldsymbol{x}$ to minimize the Lagrangian of Problem (7); [2]

$$\boldsymbol{x} := \boldsymbol{x} - \eta\nabla_{\boldsymbol{x}}\mathcal{L} = \boldsymbol{x} - \eta\boldsymbol{G}(\boldsymbol{r}\odot\boldsymbol{\alpha}). \tag{9}$$

Subsequently, given $\boldsymbol{x}$, we solve (7) to get the gradient for the next iteration. As with other gradient based methods, improvement in the primal (8) depends on how to set the parameter u and the stepsize η. At the final optimality (i.e., with all variables including $\boldsymbol{x}$), we require the following stopping criteria: $\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq \tau_\alpha,\ \|\boldsymbol{K}_r\boldsymbol{d}\|_2 \leq \tau_d$ with tolerance τ_α and τ_d.

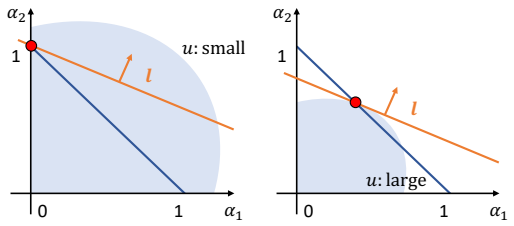


Figure 1. Illustrative example of SOC (shaded region), feasible region (blue line), objective function value (orange line), and solution (red dot) in 2-dimensional space in $\boldsymbol{\alpha}$. *Left:* too small u favors extreme points. *Right:* larger u favors intermediate solutions.

We can gain a geometric interpretation of Problem (7) by inspecting the parameter u. Figure 1 gives an illustrative example of SOC (shaded region), feasible region (blue line segment), objective function value (orange line), and solution (red dot) in 2-dimensional space in $\boldsymbol{\alpha}$. A small u tends to relax the SOC ($\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq \gamma$), and the feasible region would become large, which is depicted in Figure 1 $(left)$. In this case, the dual solution $\boldsymbol{\alpha}$ would likely to be at extreme points, which tends to gives non-smooth changes to the solutions when the solution (i.e., basis) jumps between the extreme points. The non-smooth behavior gets even worse if the stepsize is set too large for the gradient. On the other hand, if $u \gg 1$, the SOC constraint will be tightened, which pushes the solution to the cross-over point between the SOC and the convex constraint on $\boldsymbol{\alpha}$. This helps to generate smoother solutions (Figure 1 $(right)$). However, using too large u will weaken the first term in the objective function in (7) that seeks alignment with preference, which is not ideal[3]. Therefore, we need to choose the right value of u to successfully solve the entire problem.

3.3.2. Solution Strategy

As mentioned, a key to solve the problem is to find the right value of u. By choosing a small u, we seek to find a solution

[2] We do not consider $\boldsymbol{K}_r(\boldsymbol{x})$ as we would need to compute the Hessian in $\boldsymbol{x}$, which is prohibitive in $n \gg 1$ and out of scope of this paper.

[3] When $u \to \infty$, WC-MGDA reduces to the weighted version of MGDA with $\boldsymbol{K}_r$.

aligned with the preference (i.e., *WC solution* hereafter). However, we also need to minimize the norm $\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2$ for PS. This may not be achievable as evident in (7): a small u would likely yield higher value of γ, which means a looser upper bound on $\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2$. Hence, PS is not guaranteed. On the other hand, from (7), a large u tightens $\|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq \gamma$. However, a large u also means loosening $\|\boldsymbol{d}\|_2 \leq u$ – hence the 1st inequality constraint in (8), which is a key for WC. This is a worse situation than the opposite case as finding a WC solution is more difficult than finding *any* PS solution.

Although intermediate values of u may seem to be reasonable, the choice of u is suboptimal due to the competing nature of the SOC upper bounds u and γ; finding a WC solution would be slower than having a smaller u, and finding a PS solution would also be slower than having a larger u. Moreover, having a hyperparameter u requires exploration and is cumbersome for large-scale problems such as DNN models, where one model training is costly. Therefore, we propose a strategy to automatically tune u to discover a WC solution that is also PS. The high-level idea is to prioritize finding WC solutions. Once one WC solution is found, we can increase u to push the WC solution to PS for convergence. However, this is not straightforward as we need certain criterion to tell if a solution is found in one way or the other. To realize the idea, we propose the following problem[4], which is a variant of (8):

$$\min_{u,\rho,\boldsymbol{d}\in\mathbb{R}^m} \; u \;\; s.t. \; \boldsymbol{r}\odot\boldsymbol{l}(\boldsymbol{x})+\boldsymbol{K}_r\boldsymbol{d} \leq \rho\boldsymbol{e}, \|\boldsymbol{d}\|_2 \leq u, \; \rho \leq \rho^0, \tag{10}$$

where ρ^0 is the objective value in the previous iteration and constant. Namely, we use the primal objective function ρ in (8) as the criterion. As noted, the primary objective value is not guaranteed to be improved due to the sub-optimal setting of u and stepsize η. By leveraging the change in the objective function, we identify if we need to adjust u and how much to ensure monotonic non-increase of the primary objective. We have some important results on solving (10).

Lemma 3.2. *For any non-zero matrix $\boldsymbol{K}_r$, the following statements regarding feasibility of (10), Pareto stationarity, and feasibility of WC problem (5) with $\rho = \rho^0$ hold true.*

Lemma 3.2.1. If Pareto stationarity does not hold, or WC problem is feasible with $\rho = \rho^0$, then (10) is feasible.

Lemma 3.2.2. If (10) is infeasible, Pareto stationarity holds, and WC with $\rho = \rho^0$ is infeasible.

See A.1 for the proof. Intuitively, Lemma 3.2.1 covers two directions for improvement: WC problem is feasible, but not optimal (i.e., smaller ρ can be obtained), or PS is not met. The following lemma covers two possible cases when (10) is feasible.

[4]We will refrain from using $\boldsymbol{x}$ as a variable from now on as indicated in (9).

Lemma 3.3. *Suppose (10) is feasible. Either of following two cases holds*

1. *$\rho < \rho^0$, which indicates there is room for strictly decreasing the primary objective. (10) gives a trivial solution.*

2. *$\rho = \rho^0$, which means there is no room for improving ρ, and has non-trivial solutions u. The solution of (8) can be obtained from (10).*

See A.2 for the proof. Lemma 3.3 ensures solving (10) is enough to generate the gradient when there is no improvement in the primal objective. Otherwise, we need to switch to solving (7). When we find ρ strictly smaller than ρ^0, we need to directly solve (7) with current u to find a better solution since (10) yields only a trivial solution (i.e., $\boldsymbol{\alpha} = \boldsymbol{0}$). When we meet equality $\rho = \rho^0$, we find a minimum u within the constraint. This means, we prioritize WC optimization by choosing the smallest u as small u favors WC optimization. During the iterations, u may go up or down. Discovery of u smaller than the current u means we are proceeding toward WC optimization, and finding a larger u than the current u implies we cannot find a solution for further optimizing WC. At that moment, we will start increasing u to ensure PS. If it successfully finds both WC and PS, we can stop on meeting the stopping criteria. Otherwise, since a larger u means relaxation of $\|\boldsymbol{K}_r\boldsymbol{d}\|_2 \leq u$, $\|\boldsymbol{K}_r\boldsymbol{d}\|_2$ may increase, which leads to worsening of WC. This, in turn, gives us room to find smaller u and improve WC. Note that the exploration is controlled to keep the primary objective unchanged over iterations if not improved.

Lastly, suppose (10) is infeasible (i.e., Lemma 3.2.2), which means at least PS holds. In this case, we need to carefully choose u to avoid drastically deviating from the PS that is already satisfied. One can choose the same u previously used, or minimally relax (10) to reach feasibility. In our implementation, we choose to take the former approach.

3.3.3. Extended WC-MGDA

We propose Extended WC-MGDA (XWC-MGDA) by applying the following two modifications to (8).

First, as discussed in Section 1, one motivation to develop XWC-MGDA is to enable us to explore a specific portion of PF pivoted on a reference model. The reference model could be any baseline model such as an existing production model, or a pre-trained model. To this end, instead of minimizing $\boldsymbol{l}(\boldsymbol{x})$, we minimize $\boldsymbol{l}(\boldsymbol{x}) - \tilde{\boldsymbol{b}}$, where $\tilde{\boldsymbol{b}}$ is the loss function value of the reference model. By the methodology, we aim to build models that are closer or better than the reference model. If we achieve a dominating solution w.r.t. the reference model, we should have $\boldsymbol{l}(\boldsymbol{x}) - \tilde{\boldsymbol{b}} < \boldsymbol{0}$.

Second, WC method is known to generate weak PO (Kaisa, 1999), which suggests that we can have a solution with at least one objective that is not strictly better than its dominating solutions. To avoid weak PO points, traditionally, one can add a linear summation of loss functions in the objective (Kaisa, 1999), i.e., with a constant vector $\boldsymbol{v} > \mathbf{0}$, (5) can be modified as follows;

$$\min_{\rho} \ \rho + \boldsymbol{v}^\intercal(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})) \quad \text{s.t. } \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) \leq \rho \boldsymbol{e}. \tag{11}$$

The following lemma states that (11) is identical to setting a lower bound constraint on $\boldsymbol{\alpha}$.

Lemma 3.4. *The dual of* (11) *can be formulated as*

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \ \boldsymbol{\alpha}^\intercal(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})) \ s.t. \ \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1, \ \boldsymbol{\alpha} \geq \boldsymbol{w}, \ \boldsymbol{K}_r \boldsymbol{\alpha} = \mathbf{0}, \tag{12}$$

where we define $\boldsymbol{w} \equiv \boldsymbol{v}/(1 + \boldsymbol{e}^\intercal \boldsymbol{v})$*, which implies that* $\boldsymbol{w}$ *is restricted to* $\boldsymbol{e}^\intercal \boldsymbol{w} < 1$.

See A.3 for the proof. Now, we can follow derivations in Section 3.3.1 to derive the dual and primal of XWC-MGDA:

$$\begin{aligned} \max_{\gamma, \boldsymbol{\alpha} \in \mathbb{R}^m} \quad & \boldsymbol{\alpha}^\intercal(\boldsymbol{r} \odot (\boldsymbol{l}(\boldsymbol{x}) - \tilde{\boldsymbol{b}})) - u\gamma \\ s.t. \quad & \boldsymbol{e}^\intercal \boldsymbol{\alpha} = 1, \ \boldsymbol{\alpha} \geq \boldsymbol{w}, \ \|\boldsymbol{K}_r \boldsymbol{\alpha}\|_2 \leq \gamma, \end{aligned} \tag{13a}$$

$$\begin{aligned} \min_{\rho, \boldsymbol{d} \in \mathbb{R}^m, \boldsymbol{z} \in \mathbb{R}^m_+} \quad & \rho - \boldsymbol{w}^\intercal \boldsymbol{z} \\ s.t. \quad & \boldsymbol{r} \odot (\boldsymbol{l}(\boldsymbol{x}) - \tilde{\boldsymbol{b}}) + \boldsymbol{K}_r \boldsymbol{d} + \boldsymbol{z} \leq \rho \boldsymbol{e}, \\ & \|\boldsymbol{d}\|_2 \leq u. \end{aligned} \tag{13b}$$

(13) can be seen as a specialized problem to the KKT stationarity problem used in Lin et al. (2019a), and gives interpretation of the lower bound $\boldsymbol{w}$. Further, the auto-adjustment of u is computed by

$$\begin{aligned} \min_{u, \rho, \boldsymbol{d} \in \mathbb{R}^m, \boldsymbol{z} \in \mathbb{R}^m_+} \quad & u \\ s.t. \quad & \boldsymbol{r} \odot (\boldsymbol{l}(\boldsymbol{x}) - \tilde{\boldsymbol{b}}) + \boldsymbol{K}_r \boldsymbol{d} + \boldsymbol{z} \leq \rho \boldsymbol{e}, \\ & \|\boldsymbol{d}\|_2 \leq u, \ \rho - \boldsymbol{w}^\intercal \boldsymbol{z} \leq p^0, \end{aligned} \tag{14}$$

where p^0 is the primary objective function value of the previous iteration. The arguments and properties developed for the algorithm of WC-MGDA also applies to XWC-MGDA.

Note that **WC-MGDA is a special case of XWC-MGDA** with $\tilde{\boldsymbol{b}}$ and $\boldsymbol{w}$ are zero. We show the algorithm for XWC-MGDA in Algorithm 1, which subsumes that for WC-MGDA.

3.3.4. Computational complexity

Computational complexity of Algorithm 1 is $O(m^2 n)$ for $n \gg m$, which is usually the case for machine learning applications. It is dominated by the matrix computation of $\boldsymbol{G}^\intercal \boldsymbol{G}$: $O(m^2 n)$. The computation of $\boldsymbol{K}_r$ takes $O(m^3)$ due to the matrix square root calculation (Frommer & Hashemi, 2010) and SOCP takes $O(m^{2.87})$ (Kerenidis et al., 2021).

Algorithm 1 XWC-MGDA

input: preference vector $\boldsymbol{r}$, lower bound on $\boldsymbol{\alpha}$: $\boldsymbol{w}$, loss function value of a reference model $\tilde{\boldsymbol{b}}$, step size η, max. #iterations I_M, tolerance τ_α and τ_d.
randomly generate $\boldsymbol{x}_1$, initialize $p^0 \gg 1$ and $u \ll 1$
for $i = 1$ **to** I_M **do**
  compute gradient matrix $\boldsymbol{G}(\boldsymbol{x}_i)$, $\boldsymbol{K}_r(\boldsymbol{x}_i)$
  if (14) has non-trivial solution **then**
    update u
  else
    get $(\boldsymbol{\alpha}, \boldsymbol{d}, p)$ by solving (13), where p is the resulting objective function value
  end if
  if $\|\boldsymbol{K}\boldsymbol{\alpha}\|_2 \leq \tau_\alpha$ and $\|\boldsymbol{K}_r \boldsymbol{d}\|_2 \leq \tau_d$ **then**
    return: $\boldsymbol{x}_i$
  end if
  compute gradient $\boldsymbol{g}(\boldsymbol{x}_i) = \boldsymbol{G}(\boldsymbol{x}_i)(\boldsymbol{r} \odot \boldsymbol{\alpha})$. update $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \eta \boldsymbol{g}(\boldsymbol{x}_i)$, and $p^0 = p$
end for

4. Experiments

4.1. Synthetic data

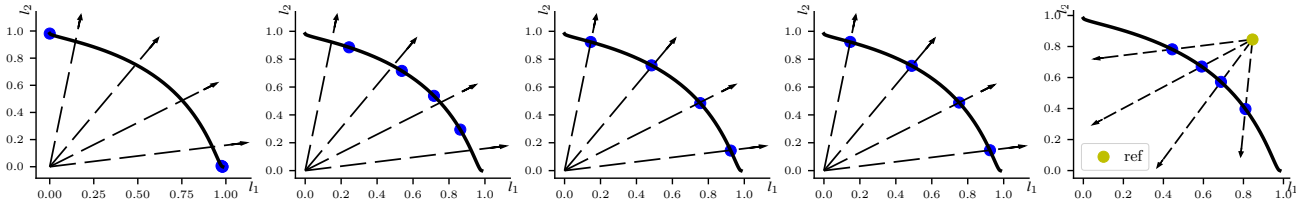
We illustrate the behavior of the algorithms on synthetic dataset from state-of-the-art papers (Lin et al., 2019b; Mahapatra & Rajan, 2020):

$$\boldsymbol{l}(\boldsymbol{x}) = \left[1 - e^{-\|\boldsymbol{x} - \frac{1}{\sqrt{n}}\boldsymbol{e}\|_2^2}, \ 1 - e^{-\|\boldsymbol{x} + \frac{1}{\sqrt{n}}\boldsymbol{e}\|_2^2}\right]^\intercal, \tag{15}$$

which is known to have highly non-convex PF. We use the dimensionality $n = 20$ and the same initial point across all models generated by uniform random sampling. We compare linear scalarization (LinScalar), i.e., fixed linear combination of objectives / tasks, PMTL (Lin et al., 2019b), and EPO (Mahapatra & Rajan, 2020), WC-MGDA and XWC-MGDA with a reference point (RP).

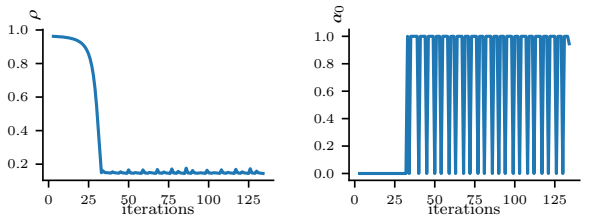
Figure 2 shows the results with several preference vectors generated by equiangular directions from the origin for LinScalar, PMTL, EPO, WC-MGDA, and those from a RP with XWC-MGDA. We can gain the following observations;

- LinScalar is only able to find the convex part of the PF.
- PMTL can find solutions roughly aligned with the preferences due to the use of them to divide the objective space into subregions. However, the alignments are not perfect.

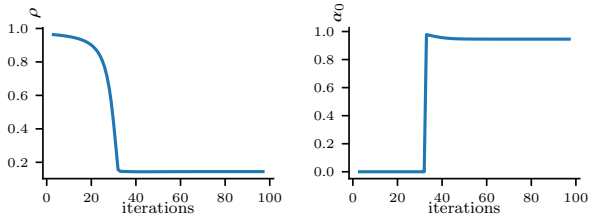


(a) LinScalar (b) PMTL (c) EPO (d) WC-MGDA (e) XWC-MGDA

Figure 2. Comparison of various methods. Dashed arrows represent preference directions.



(a) ρ for $u = 0.001$ (b) α for $u = 0.001$



(c) ρ for auto-tune (d) α for auto-tune

Figure 3. The effect of auto adjustment. A small value of u yields non-smooth behavior in (a) and (b). Auto-tuning helps stabilize it, resulting in a faster convergence as shown in (c) and (d), for both the primary objective ρ and solution α.

- EPO and WC-MGDA can achieve significantly better alignments compared with PMTL.

- Further, once a RP is given, XWC-MGDA is able to explore PF that dominates the RP, which is also aligned with the preference. **This is the distinctive advandange of XWC-MGDA over other five methods.**

Next, we illustrate the effect of the auto-tuning of u. In Figure 3, we plot the primal objective function ρ and α_0 by iterations with WC-MGDA. Figure 3 (a) and (b) are from the fixed policy of u with a small value 0.001, which results in slower convergence (greater than 130 iterations). Figure 3 (c) and (d) are from the auto tuning logic. As illustrated, setting a small u value introduces increase in objective function ρ and non-smooth behavior of the solution α_0. The auto adjustment of u ensures monotonic non-increase of ρ and smoother / faster convergence to the desired solution (smaller than 100 iterations).

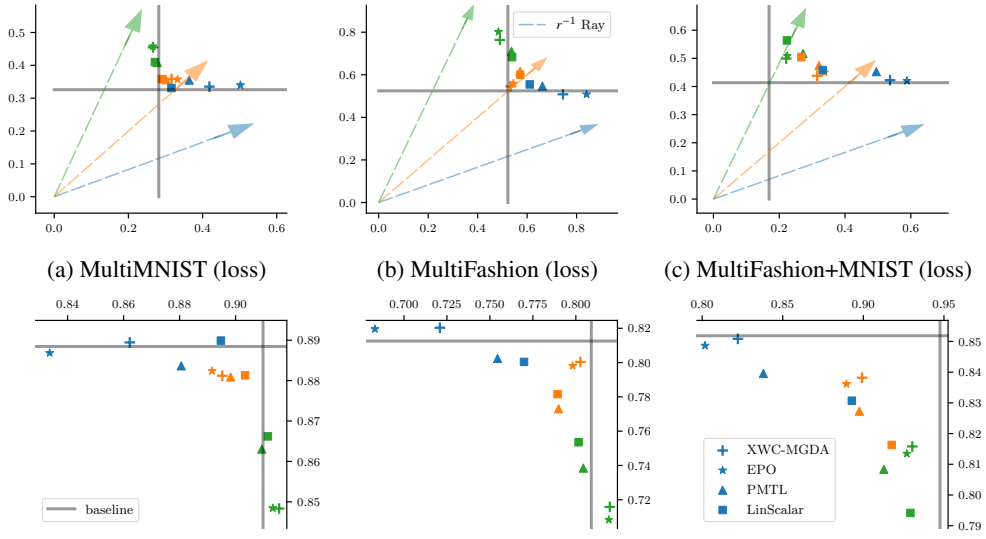
4.2. Real dataset

4.2.1. Image classification

For image classification, we use three datasets: (1) Multi-MNIST (Sabour et al., 2017), (2) Multi-Fashion (Xiao et al., 2017), and (3) Multi-Fashion+MNIST (Lin et al., 2019b). These are the same datasets used by Lin et al. (2019b) and construction processes are detailed in Lin et al. (2019b). In each dataset, there are 120,000 samples in the training set and 20,000 samples in the test set. For each dataset, we have two tasks; 1) classifying the top-left image, and 2) classifying the bottom-right image. For a fair comparison, we apply LeNet (LeCun et al., 1998) used in (Lin et al., 2019b; Mahapatra & Rajan, 2020) as the MTL neural network. The baseline for comparison is training the network for individual tasks. Furthermore, to reduce the variability, we use the same random seed across different methods. Note we apply XWC-MGDA without RP when comparing with other methods. We use $\boldsymbol{w} = 10^{-6}\boldsymbol{e}$ for all cases.

First, we test the performance of all methods given three preference vectors. Ideal solutions should lie on these rays (i.e., inverse of preference), or dominate others. The top row in Figure 4 shows test losses of the methods and bottom row shows test accuracies. In terms of losses, XWC-MGDA is able to generate models aligned with the preference, especially for green and yellow rays. For blue rays, EPO seems to be closer to the preference. However, XWC-MGDA either dominates other methods or achieves similar performance. While EPO seeks weak PO, XWC-MGDA obtains PO, which means that if PF looks flat in Task 2, it tries to find one that is better in Task 1. The results seem to show such a behavior, which can be confirmed from the accuracy where XWC-MGDA tends to dominate EPO as well as other gradient methods.

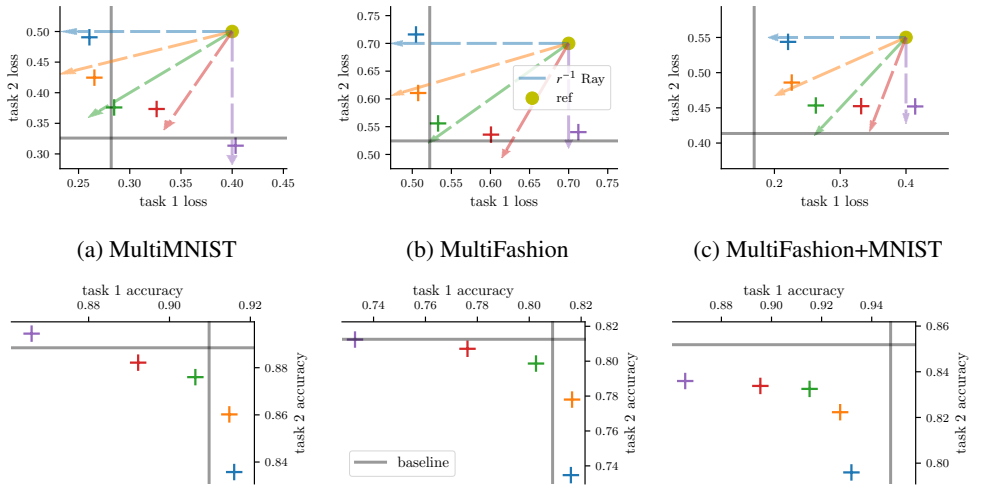
Figure 5 illustrates how XWC-MGDA (with RP) performs on these three datasets. As shown in the figure, the models are well aligned with rays originating from the given RP. This means that it finds better solutions than the reference with a given preference. This yields a significant reduction of exploration to find a model that at least matches all losses / objectives from $\Omega(m)$ (Momma et al., 2020) to $O(1)$. The resulting PF for both cases (i.e., with or without RP) is



(a) MultiMNIST (loss) (b) MultiFashion (loss) (c) MultiFashion+MNIST (loss)

(d) MultiMNIST (accuracy) (e) MultiFashion (accuracy) (f) MultiFashion+MNIST (accuracy)

Figure 4. Comparison of methods with preference rays from the origin. X-axis and Y-axis are for task 1 and 2, respectively.



(a) MultiMNIST (b) MultiFashion (c) MultiFashion+MNIST

(d) MultiMNISTA (e) MultiFashion (f) MultiFashion+MNIST

Figure 5. XWC-MGDA with reference point $(0.5, 0.5)$, $(0.7, 0.7)$, and $(0.55, 0.55)$ for three datasets, respectively.

Table 2. Hypervolumes for experiments in Section 4.2

	XWC-MGDA	EPO	PMTL	LinScalar
MultiMNIST Loss	0.0142	0.0135	0.0141	**0.0168**
MultiFashion Loss	**0.0677**	0.0602	0.0590	0.0675
MultiFashion+MNIST Loss	**0.0389**	0.0376	0.0303	0.0363
MultiMNIST Accuracy	0.0016	0.0015	0.0015	**0.0019**
MultiFashion Accuracy	**0.0099**	0.0087	0.0087	**0.0099**
MultiFashion+MNIST Accuracy	**0.0052**	0.0050	0.0040	0.0048
River Loss	**6.83E+28**	5.93E+28	1.08E+28	6.64E+28
Emotion Loss	0.000348	**0.000366**	0.000230	0.000258

similar to each other.

4.2.2. Multi-target regression

We use the multi-target regression dataset, i.e., River Flow dataset (Spyromitros-Xioufis et al., 2016), that considers the prediction of river network flows for 48 hours in the future at 8 sites in the Mississippi River network. We note that it has also been used in Mahapatra & Rajan (2020). To keep consistency, we take the same data processing steps, use the same fully connected feed-forward neural network (FNN) with 4 layers, and Mean Squared Error (MSE) as the loss for each task. We train the FNN using XWC-MGDA, EPO search, PMTL and LinScalar. We generate 20 preference vectors.

We report the results using the relative loss profile (RLP) $\boldsymbol{r} \odot \boldsymbol{l}$ on the test data in Figure 6. We notice that XWC-MGDA and EPO perform quite similarly and are the best among these methods; RLPs of both methods distribute more uniformly across the 8 tasks than PMTL and Linear Scalarization. Further, losses for all 8 tasks are similar to those of Baseline and are much smaller than the other multi-task learning methods, especially PMTL.

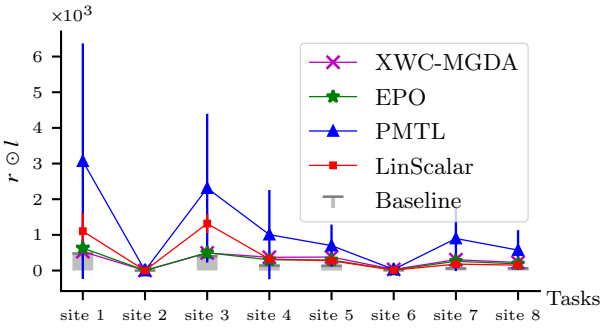


Figure 6. Multi-target regression experiment. RLP (mean and standard deviation) is plotted for each task. XWC-MGDA and EPO show superior performance than others.

4.2.3. Multi-class classification

We next use the multi-class classification dataset, i.e., Emotions and Music (Trohidis et al., 2011), as a counterexample to show when MOO methods would fail to outperform LinScalar and Baseline. We consider the prediction of 6 emotions among a set of 593 songs based on the Tellegen-Watson-Clark model of affect. To keep it consistent with Mahapatra & Rajan (2020), we take the same data processing steps, use fully connected feed-forward neural network (FNN) with 4 layers, and Sigmoid Binary Cross Entropy (SBCE) as the loss for each task. We train the FNN using WC-MGDA, EPO, PMTL and LinScalar. We generate 50 preference vectors. Again, we report results using RLP on the test data in Figure 7. We notice that all of the methods have almost uniform RLPs and none of them dominate each other, which is expected as the PF might be convex as noticed in Mahapatra & Rajan (2020).

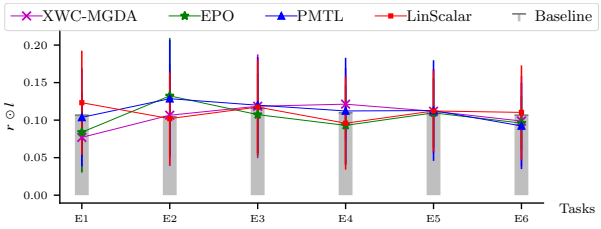


Figure 7. Multi-class classification experiment. RLP (mean and standard deviation) is plotted for each task. LinScalar works well, implying that the shape of PF might be convex.

4.2.4. Hypervolumes for all experiments

To quantify the performance of these methods, we report the mean value of hypervolumes (HV) for all experiments in Section 4.2 as shown in Table 2. For image classification, we repeat five trials, using various random seeds that are shared across all methods. We use Nadir points[5] as the reference points when computing HVs. We see that XWC-MGDA achieves the highest HV in five out of eight results (second best for the remaining three), indicating its superior capability of generating strong PF that dominates others.

5. Conclusion

In this paper, we developed a novel and generic framework to discover a Pareto optimal (PO) solution with multiple forms of preference. It allows us to formulate a generic MOO / MTL problem to express a preference, which is solved to achieve the PO that is aligned with the preference. Specifically, we applied the framework to solve the weighted Chebyshev problem using WC-MGDA, and an extended weighted Chebyshev problem using XWC-MGDA. While WC-MGDA solves a problem similar to existing methods such as EPO, XWC-MGDA can further explore PF from any given reference point. This means we can build a model that is similar to or better than the reference (e.g., existing / pre-trained) model with only one attempt, which yields a significant reduction of exploration from $\Omega(m)$ in (Momma et al., 2020) to $O(1)$. The framework is generic and opens up a door for incorporating preferences in a variety of ways such as ϵ-constraint method (Kaisa, 1999) that imposes upper bounds on losses to express a hard constraint preference (i.e., we do not want to sacrifice), and even a combination of XWC and ϵ-constraint method. Experimental results demonstrated the method achieves competitive performance with existing methods, and the performance can be achieved from different forms of preferences (i.e., XWC-MGDA with or without a reference point).

[5] Worst performance on single task baselines

References

Alizadeh, F. and Goldfarb, D. Second-order cone programming. *MATHEMATICAL PROGRAMMING*, 2001.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787.

Carmel, D., Haramaty, E., Lazerson, A., and Lewin-Eytan, L. Multi-objective ranking optimization for product search using stochastic label aggregation. In *Proceedings of The Web Conference*, 2020.

Caruana, R. Multitask learning. *Machine learning*, 1997.

Désidéri, J.-A. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350:313–318, 03 2012.

Fliege, J. and Svaiter, B. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 2000.

Frommer, A. and Hashemi, B. Verified computation of square roots of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 2010.

Gong, C., Liu, X., and Liu, Q. Automatic and harmless regularization with constrained and lexicographic optimization: A dynamic barrier approach. In *Advances in Neural Information Processing Systems*, 2021.

Kaisa, M. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, USA, 1999.

Kerenidis, I., Prakash, A., and Szilágyi, D. Quantum algorithms for Second-Order Cone Programming and Support Vector Machines. *Quantum*, 2021. doi: 10.22331/q-2021-04-08-427.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Lin, X., Chen, H., Pei, C., Sun, F., Xiao, X., Sun, H., Zhang, Y., Ou, W., and Jiang, P. A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019a.

Lin, X., Zhen, H.-L., Li, Z., Zhang, Q.-F., and Kwong, S. Pareto multi-task learning. In *Advances in Neural Information Processing Systems 32*. 2019b.

Lin, X., Yang, Z., Zhang, Q., and Kwong, S. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020.

Liu, X., Tong, X., and Liu, Q. Profiling pareto front with multi-objective stein variational gradient descent. *Advances in Neural Information Processing Systems*, 2021.

Ma, P., Du, T., and Matusik, W. Efficient continuous pareto exploration in multi-task learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Mahapatra, D. and Rajan, V. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Matouek, J. and Gärtner, B. *Understanding and Using Linear Programming (Universitext)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3540306978.

Momma, M., Garakani, A. B., and Sun, Y. Multi-objective relevance ranking. In *Proceedings of the SIGIR 2019 Workshop on eCommerce*, volume 2410 of *CEUR Workshop Proceedings (preprint)*. CEUR-WS.org, 2019.

Momma, M., Garakani, A. B., Ma, N., and Sun, Y. Multi-objective ranking via constrained optimization. In *Companion Proceedings of the The Web Conference*, 2020.

Navon, A., Shamsian, A., Fetaya, E., and Chechik, G. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations*, 2021.

Ruchte, M. and Grabocka, J. Scalable pareto front approximation for deep multi-objective learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2021.

Ruder, S. An overview of multi-task learning in deep neural networks, 2017.

Sabour, S., Frosst, N., and Hinton, G. E. In *Advances in Neural Information Processing Systems*, 2017.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31*. 2018.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 2016.

Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.

Zhang, Q. and Li, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 2007.

A. Proofs

A.1. Proof of Lemma 3.2

Preliminaries.
First, let us prepare several materials for the proof. As our interest is feasibility, we can ignore the constraint on $\|\boldsymbol{d}\|_2$: $\|\boldsymbol{d}\|_2 \leq u$, since we can take arbitrarily large u. Using $\rho \leq \rho^0$, we can rewrite the first constraint in (10):

$$\boldsymbol{K}_r\boldsymbol{d} \leq \rho^0\boldsymbol{e} - \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) \equiv \boldsymbol{s}^0. \tag{16}$$

Note the sign of i-th element of $\boldsymbol{s}^0$ indicates feasibility of the original WC problem: (5) is feasible if and only if $\boldsymbol{s}^0 \geq 0$, and (5) is infeasible if and only if $\exists i, s.t., s_i^0 < 0$.

Proof. Let us prove Lemma 3.2.2 first.

By applying the variant of Farkas' lemma (Matouek & Gärtner, 2006) to (16), we know exactly one of the followings holds true;

(i) (16) holds true (i.e., (10) is feasible)

(ii) $\boldsymbol{K}_r\boldsymbol{\alpha} = \boldsymbol{0}$, $\boldsymbol{\alpha} \geq \boldsymbol{0}$, and $\boldsymbol{s}^{0\intercal}\boldsymbol{\alpha} < \boldsymbol{0}$.

Suppose (10) is infeasible, which means (i) does not hold, and (ii) must hold. We have

$$\boldsymbol{K}_r\boldsymbol{\alpha} = \boldsymbol{0}, \boldsymbol{\alpha} \geq \boldsymbol{0}, \text{ and } \boldsymbol{s}^{0\intercal}\boldsymbol{\alpha} < \boldsymbol{0}. \tag{17}$$

The last inequality $\boldsymbol{s}^{0\intercal}\boldsymbol{\alpha} < \boldsymbol{0}$ implies,

(a) $\exists i, s.t., \alpha_i > 0$ (this can be readily proved by contradiction), and

(b) $s_i^0 < 0, \ \exists\, i \in \{i|\alpha_i > 0\}$.

From (a), by the existence of non-zero α_i, we can rescale $\boldsymbol{\alpha}$ to derive the Pareto stationarity condition:

$$\boldsymbol{K}_r\tilde{\boldsymbol{\alpha}} = 0, \tilde{\boldsymbol{\alpha}} \geq \boldsymbol{0}, \boldsymbol{e}^\intercal\tilde{\boldsymbol{\alpha}} = 1, \tag{18}$$

with $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}/\boldsymbol{e}^\intercal\boldsymbol{\alpha}$. Further, as discussed in *Preliminaries*, the existence of strictly negative s_i^0 implies WC is infeasible with $\rho = \rho^0$. Therefore, (ii) implies both Pareto stationarity and WC infeasibility must be true, which concludes Lemma 3.2.2.

For Lemma 3.2.1, we simply take contraposition of Lemma 3.2.2, which concludes Lemma 3.2.1. □

A.2. Proof of Lemma 3.3

Proof. The Lagrangian of (10) is given as follows;

$$\mathcal{L} = u + \boldsymbol{\alpha}^\intercal(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{K}_r\boldsymbol{d} - \rho\boldsymbol{e}) + \boldsymbol{\beta}^\intercal\boldsymbol{d} - \gamma u + \delta(\rho - \rho^0), \tag{19}$$

where $\boldsymbol{\alpha}$, $(\boldsymbol{\beta}, \gamma) \in \mathcal{C}$, and δ are dual variables associated with inequalities and SOC in (10). By complementarity, $\rho < \rho^0$ implies $\delta = 0$. However, by the stationarity condition $\partial\mathcal{L}/\partial\rho = 0$, we have $\boldsymbol{e}^\intercal\boldsymbol{\alpha} = \delta$, i.e., $\boldsymbol{e}^\intercal\boldsymbol{\alpha} = 0$. Further, since $\boldsymbol{\alpha} \geq 0$, $\boldsymbol{\alpha} = \boldsymbol{0}$ must be met, which is a trivial solution and proves the 1st part of the lemma.

If $\rho = \rho^0$, δ can take a finite value and hence $\boldsymbol{\alpha}$. By using stationarity conditions $\nabla_{\boldsymbol{d}}\mathcal{L} = \boldsymbol{0}$ and $\partial\mathcal{L}/\partial u = 0$, the dual of (10) is derived as follows;

$$\max \ \boldsymbol{\alpha}^\intercal\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) - \delta\rho^0 \ \text{ s.t. } \ \boldsymbol{\alpha}^\intercal\boldsymbol{e} = \delta, \ \|\boldsymbol{K}_r\boldsymbol{\alpha}\|_2 \leq 1. \tag{20}$$

Assume all KKT conditions are met at the optimality. Let $(\tilde{\boldsymbol{\alpha}}, \tilde{\delta}, \tilde{\rho})$ be the solution for the dual (20), and $(\tilde{u}, \tilde{\boldsymbol{d}}, \tilde{\rho})$ be the solution for the primal (10). Let us define the following rescaling:

$$\tilde{\boldsymbol{\alpha}}_2 = \tilde{\boldsymbol{\alpha}}/\tilde{\delta}, \ \tilde{\gamma}_2 = 1/\tilde{\delta}. \tag{21}$$

By applying (21) to (20), we get

$$\max\ \delta(\tilde{\boldsymbol{\alpha}}_2^\intercal \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) - \rho^0)\ \text{ s.t. }\ \tilde{\boldsymbol{\alpha}}_2^\intercal \boldsymbol{e} = 1,\ \|\boldsymbol{K}_r \tilde{\boldsymbol{\alpha}}_2\|_2 \leq \gamma_2. \tag{22}$$

At the optimality, due to the strong duality, the objective value must be equal to the primal objective value, i.e., $\tilde{u}$:

$$\tilde{u} = \delta(\tilde{\boldsymbol{\alpha}}_2^\intercal \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) - \rho^0) \tag{23}$$

Let $(\bar{\boldsymbol{\alpha}}, \bar{\gamma})$ be the solution of (7). By plugging $\tilde{u}$ into the objective function in (7), we get

$$\bar{\boldsymbol{\alpha}}^\intercal \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) - \tilde{u}\bar{\gamma} = (\bar{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}}_2)^\intercal \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) + \rho^0 \bar{\gamma}/\tilde{\gamma}_2 \tag{24}$$

By the precondition $\rho = \rho_0$ at the optimality in (10), we have

$$(\bar{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\alpha}}_2)^\intercal \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) + \rho^0 \bar{\gamma}/\tilde{\gamma}_2 \equiv \rho^0. \tag{25}$$

Since the identity equation (25) must hold for any $(\tilde{\boldsymbol{\alpha}}_2, \tilde{\gamma}_2)$, we conclude $\tilde{\boldsymbol{\alpha}}_2 = \bar{\boldsymbol{\alpha}}$, $\tilde{\gamma}_2 = \bar{\gamma}$, which implies solution of (7) can be obtained by (10) by applying the rescaling (21).

□

A.3. Proof of Lemma 3.4

Proof. Using $\tilde{\boldsymbol{\alpha}}$ as the dual variable for the inequality constraint of (11), the Lagrangian is given by

$$\mathcal{L} = \rho + \boldsymbol{v}^\intercal(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})) + \tilde{\boldsymbol{\alpha}}(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) - \rho \boldsymbol{e}). \tag{26}$$

From the stationarity condition on ρ: $\partial\mathcal{L}/\partial\rho = 0$, we have $\boldsymbol{e}^\intercal\tilde{\boldsymbol{\alpha}} = 1$. Further, from the stationarity condition on $\boldsymbol{x}$: $\nabla_{\boldsymbol{x}}\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{0}$ and definition of $\boldsymbol{K}_r$, we have $\boldsymbol{K}_r(\tilde{\boldsymbol{\alpha}} + \boldsymbol{v}) = \boldsymbol{0}$. Let $\tilde{\tilde{\boldsymbol{\alpha}}} \equiv \tilde{\boldsymbol{\alpha}} + \boldsymbol{v}$. As $\boldsymbol{e}^\intercal\tilde{\boldsymbol{\alpha}} = 1$, we have $\boldsymbol{e}^\intercal\tilde{\tilde{\boldsymbol{\alpha}}} = 1 + \boldsymbol{e}^\intercal\boldsymbol{v}$. By rescaling: $\boldsymbol{\alpha} \equiv 1/(1 + \boldsymbol{e}^\intercal\boldsymbol{v})\tilde{\tilde{\boldsymbol{\alpha}}}$, we have

$$\boldsymbol{K}_r\boldsymbol{\alpha} = \boldsymbol{0},\ \boldsymbol{e}^\intercal\boldsymbol{\alpha} = 1. \tag{27}$$

Since the dual vector is non-negative: $\tilde{\boldsymbol{\alpha}} \geq \boldsymbol{0}$, $\tilde{\tilde{\boldsymbol{\alpha}}} = \tilde{\boldsymbol{\alpha}} + \boldsymbol{v}$ implies $\tilde{\tilde{\boldsymbol{\alpha}}} \geq \boldsymbol{v}$. By the definition of $\boldsymbol{\alpha}$, we have

$$\boldsymbol{\alpha} = \frac{1}{1 + \boldsymbol{e}^\intercal\boldsymbol{v}}\tilde{\tilde{\boldsymbol{\alpha}}} \geq \frac{1}{1 + \boldsymbol{e}^\intercal\boldsymbol{v}}\boldsymbol{v} = \boldsymbol{w}. \tag{28}$$

Note the last equality is by the definition of $\boldsymbol{w}$. Now, using (28), it is readily shown

$$\boldsymbol{e}^\intercal\boldsymbol{w} = \boldsymbol{e}^\intercal\boldsymbol{v}/(1 + \boldsymbol{e}^\intercal\boldsymbol{v}) < 1. \tag{29}$$

Finally, plugging in all results into Eq. (26) and ignoring constant terms, we have

$$\max\ \boldsymbol{\alpha}^\intercal(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}))\ \text{s.t.}\ \boldsymbol{e}^\intercal\boldsymbol{\alpha} = 1,\ \boldsymbol{\alpha} \geq \boldsymbol{w},\ \boldsymbol{K}_r\boldsymbol{\alpha} = \boldsymbol{0}, \tag{30}$$

which is equivalent to (12). □