Pareto-Informed Smart "Predict, then Optimize" for Multi-Objective Combinatorial Problems

Aabhash Dhakal[a,*] **and Tim Lachner**[a]

[a]University of Würzburg
ORCID (Aabhash Dhakal): https://orcid.org/0009-0006-1245-6824, ORCID (Tim Lachner): https://orcid.org/0009-0001-6982-1697

Abstract. Many real-world planning applications, from logistics, routing to resource allocation, give rise to optimization problems that need to balance multiple, conflicting objectives. While machine learning is often used to predict the unknown parameters or costs of these problems, standard models trained on prediction accuracy can lead to suboptimal plans. Decision-Focused Learning (DFL) addresses this by integrating the downstream optimization task into the training objective. However, existing DFL methods for multi-objective problems rely on simple scalarization, which fails to capture the rich trade-offs of the Pareto Frontier. To address this gap, we propose a Pareto-aware extension to single-objective Smart "Predict, then Optimize" (SPO+) for multi-objective combinatorial optimization problems. Our loss function leverages the true Pareto Frontier, forming a target decision that aligns with a sampled preference vector, and minimizes regret against this target. Experiments on multi-objective shortest path and multi-objective knapsack problems show that our approach outperforms traditional two-stage Mean Squared Error (MSE) baselines. It also surpasses scalarization-based SPO+ approaches.

1 Introduction

Decision making in complex operational domains, such as logistics, infrastructure management, or resource allocation, frequently involves navigating complex trade-offs between multiple conflicting objectives [9, 13, 15]. For example, a logistics provider must constantly balance the speed of delivery against fuel consumption and monetary cost, where the optimal choice is rarely the same for all three objectives. For problems with a discrete solution space, Multi-Objective Combinatorial Optimization (MOCO) [1, 11] provides a formal framework for finding optimal trade-offs; however, this requires the objective coefficients (i.e. costs) to be known in advance.

A fundamental difficulty in applying automated planning to real-world domains is that these crucial parameters are often uncertain and dynamic. In a transportation network, the cost to traverse a segment of a road is not a static value, but a function of time-varying characteristics such as current weather patterns, traffic congestion, time of day or whether it is a holiday [21]. This uncertainty has led to the widespread adoption of "predict, then optimize" approach [18, 26]. In this two-stage approach, a machine learning (ML) model first predicts the unknown cost vector, $\hat{c}$, based on observable features, x. Subsequently, these predictions are treated as fixed parameters which are fed into a deterministic combinatorial solver to compute a supposedly optimal plan or decision, $w(\hat{c})$.

While modular and intuitive, this sequential approach suffers from a critical weakness known as the "prediction-decision gap"[7, 8, 23]. Standard ML approaches are typically trained to maximize predictive accuracy by minimizing a loss function like Mean Squared Error (MSE) between the predicted costs, $\hat{c}$ and the true costs, c. The underlying assumption is that more accurate predictions will invariably lead to better decisions. However, a substantial body of research has demonstrated that this assumption can be false [7, 17]. The reason is that not all prediction errors are equal; a model with a low overall error may make small mistakes on costs critical to the solver, leading to highly suboptimal decisions, while a model with higher error on irrelevant costs may yield a near-optimal decision. This misalignment between the learning objective and the ultimate decision-making goal can lead to demonstrably poor outcomes in practice.

To bridge this gap, Decision-Focused Learning (DFL) has emerged as an alternative [17, 22]. Recent works, such as "Smart Predict, then Optimize" (SPO+), directly incorporate the structure of the downstream optimization problem into the ML model's training objective [7]. Instead of minimizing prediction error, they aim to minimize decision regret i.e. the quality loss incurred by making a decision based on predicted costs compared to the optimal decision that would have been made with true costs. This allows the ML model to focus on parts of the prediction that have impact on the actual solutions.

While the DFL framework has demonstrated considerable efficacy for single-objective problems its extension to multi-objective optimization contexts remains unexplored. In such problems, there is typically no single solution that is optimal across all objectives simultaneously. The concept of a single "best" solution is replaced by the concept of Pareto optimality. A solution is Pareto optimal if it is impossible to improve its performance on any single objective without degrading its performance on at least one other [12]. The set of all such solutions constitutes the Pareto set, and their corresponding values in the objective space form the Pareto Front.

A common strategy for adapting single-objective DFL to a multi-objective setting is to use linear scalarization, which combines multiple objectives into one using a preference vector. This approach is fundamentally limited, as it is only capable of finding solutions on the convex hull of the Pareto Front and fails for many combinatorial problems where the front is non-convex. This leaves a significant research gap: *the need for a DFL method that is both computation-*

* Corresponding Author. Email: aabhash.dhakal@uni-wuerzburg.de

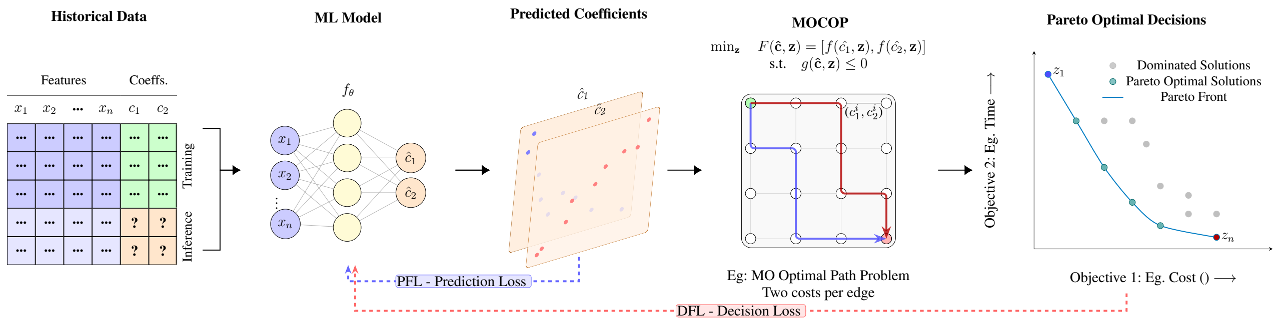

Figure 1. Data Driven Multi-Objective Combinatorial Optimization - Prediction Focused Learning vs Decision Focused Learning illustrated via a bi-objective optimization example with two set of costs

ally tractable and can effectively handle the non-convex trade-offs inherent in multi-objective problems.

In this paper, we introduce an extension to SPO+ specifically designed for multi-objective combinatorial optimization problems. While there have been previous works that extend SPO+ to Multi-Task settings [24] with multiple independent objectives, our work extends SPO+ to problems with true conflicting objectives that necessitate trade-offs. We introduce a loss function that overcomes the limitations of naive scalarization in the multi objective setting, whilst maintaining its advantages. In our approach, we inform the learning process with knowledge of the true Pareto Front. An ideal target solution from the Pareto Front is selected based on cosine similarity [19] with a sampled Preference, whilst still solving the problem via scalarization. This provides an accurate and geometrically sound learning signal to the model, guiding it to learn cost predictions that lead to high-quality decisions even in non-convex regions of the solution space.

Our Pareto-informed approach significantly outperforms standard baselines in minimizing decision regret. We demonstrate the efficacy and generality of our approach across multiple domains, with strong performance on both the Multi-Objective Shortest Path and Multi-Objective Knapsack problems [3]. Our implementation utilizes the popular PyEPO framework [25], and extends it to generate multi-objective costs, providing a valuable baseline for future research in this area.

2 Related Works

DFL methods [17], in an effort to close the "prediction-decision gap", integrate downstream optimization problem directly into the model's training process, optimizing for decision quality rather than pure predictive accuracy. Mandi et al. [17] provide a comprehensive overview of existing DFL methods, benchmarks, challenges, and possible extensions.

The seminal work in this area is the **Smart "Predict, then Optimize" (SPO+)** loss by Elmachtoub and Grigas [7]. It outperforms standard prediction-focused methods, specially under conditions of model misspecification. While the SPO loss is discontinuous and does not provide signals for gradient descent, the authors use a convex surrogate to make learning process work. They term this loss SPO+. In addition, the authors show that the SPO+ is Fisher Consistent, which means, that under full distributional knowledge and mild conditions, minimizing the surrogate loss is equivalent to minimizing true SPO loss. Likewise, they also show that with increasing sample size, decision made using SPO+ converge to full-information optimal decisions.

Tang and Khalil [24] propose an extension to Smart "Predict, then Optimize" for Multi-Task learning where decision error is minimized for multiple, related optimization tasks simultaneously. Their work differs from ours in that their objectives are independent. The tasks might share underlying information or predictions, but each is a different optimization problem. Multi-Task Loss focuses on training one model for many different problems, each with its own objective, while Multi-Objective Loss focuses on training a model for one problem that has many objectives within that single problem. Ruchte and Grabocka [20] show how Multi-Task problems are fundamentally different from Multi-objective problems.

The principles of DFL have been successfully applied to various combinatorial problems, including shortest path [7], knapsack problems [16], and energy-cost aware scheduling [16], demonstrating its broad applicability. Tang and Khalil [25] developed PyEPO, A PyTorch-based End-to-End Predict-then-Optimize Tool which provides a collection of datasets, problems and methods for solving predict-then-optimize problems with the linear objective function. We use problems and datasets from their work as inspiration for our experimental design.

Separate from the DFL paradigm, there is a rich literature on Multi-Objective Machine Learning, which focuses on training models that must balance multiple, often conflicting, loss functions. A prominent area is Pareto Multi-Task Learning (MTL). Lin et al. [14] propose decomposing the multi-objective problem into a set of constrained subproblems, each guided by a different preference vector. By solving these subproblems in parallel, their method can generate a set of diverse Pareto solutions. This approach of dedicating a separate optimization process for each preference vector is a direct inspiration for our work, where we train one specialized model for each known preference.

Likewise, Ruchte and Grabocka [19] propose COSMOS, a method that generates the entire Pareto Front from a single model in one training run. COSMOS conditions the model by augmenting the input features with a preference vector. To ensure a well-spread Pareto Front when using linear scalarization, they introduce a penalty term that maximizes the cosine similarity between the preference vector and the resulting vector of losses. The use of cosine similarity as a mechanism for aligning a solution with a preference vector was a key inspiration for our work. However, we adapt this concept for a different purpose: rather than using it as a penalty in the loss function, we use it as a heuristic to select the most relevant target solution from the true Pareto Front to guide the SPO+ gradient.

3 Problem Description and Proposed Approach

Consider a supervised learning setting for a Multi-Objective Combinatorial Optimization Problem (MOCOP). For each instance n in our data, we have a feature vector $x \in \mathcal{X}$. The goal is to predict a multi-objective cost vector $\hat{c} \in \mathbb{R}^{m \times K}$, where m is the number of decision variables (e.g., edges in a network, or selected items in the knapsack problem) and K is the number of conflicting objectives. The corresponding true cost vector is denoted by c. These costs are used in a combinatorial optimization problem defined on a set of feasible solutions $\mathcal{W}$. A solution $w \in \mathcal{W}$ is a vector representing a specific choice (e.g., a path in the shortest path problem). The quality of a solution w is evaluated by its total cost for each objective, given by the vector $z = c^T w$. The goal of MOCOP is to find the set of non-dominated solutions, known as the Pareto set $W^* \subseteq \mathcal{W}$, and its corresponding image in the objective space, the Pareto Front Z^*.

3.1 Smart "Predict, then Optimize" (SPO+)

The SPO+ loss function is a continuous and convex surrogate for the true decision regret, making it tractable for gradient-based optimization.

$$\mathcal{L}_{SPO+}(\hat{c}, c) = \max_{w \in \mathcal{W}} \{(2\hat{c} - c)^T w - (2\hat{c} - c)^T w^*\} \quad (1)$$

The gradient of SPO+ loss provides a powerful and direct learning signal. Geometrically, it represents a vector pointing from the sub-optimal solution induced by the current prediction toward the true optimal solution, w^*. By backpropagating the gradient, the learning algorithm adjusts the model's parameters in a direction that causes its future predictions to induce plans that are "closer" to the true optimum. This is a more informative signal than that provided by MSE, which only encourages the cost vector $\hat{c}$ to be numerically closer to c, without regard for which components of the cost vector are actually critical for the final decision. The SPO+ loss thus focuses the model's learning capacity on correcting prediction errors that have the largest impact on decision quality.

3.2 Multi Objective Optimization

Multi-objective optimization deals with problems that trade-off conflicting goals. The goal is not to find a single optimal solution, but rather the set of optimal trade-offs. This is formalized by the concepts of Pareto dominance and optimality [27]. A solution w_a is said to *strictly dominate* another solution w_b, denoted $w_a \succ w_b$, if it is strictly better in all objectives. A solution w_a *dominates* w_b if it is no worse than w_b in all objectives and strictly better in at least one. A solution is **Pareto optimal** if no other feasible solution dominates it. The set of all such solutions forms the Pareto set, and their corresponding values in the objective space constitute the **Pareto Front** [27, 3]. Predominant strategies exist for handling multiple objectives include:

1. **Scalarization-Based Methods:** Scalarization approaches convert the multi-objective problem into a single-objective problem. The most common technique is Linear Scalarization (LS), also termed the weighted sum method, where a user-defined preference vector $p \in \mathbb{R}_+^K$ assigns a weight to each of the k objectives, $\mathcal{L}_k(\theta)$. The problem is then reformulated as minimizing a single, scalar objective, $\min_\theta \sum_{k=1}^K p_k \mathcal{L}_k(\theta)$. LS is computationally efficient and allows direct application of single-objective solvers and DFL frameworks like SPO+. However, it can only find solutions on the *convex hull* of the Pareto Front [10]. It is guaranteed to fail for any problems where the true Front is non-convex, which is common in combinatorial optimization.
2. **Pareto-Based Methods:** In contrast, Pareto-based methods, such as Multi-Objective Evolutionary Algorithms (MOEAs) work by maintaining a population of solutions and using Pareto dominance as the selection criterion [6]. These methods are designed to approximate the entire Pareto Front, including non-convex regions. However, they are often more computationally intensive and are not easily integrated into end-to-end, gradient-based learning frameworks. While these methods are more powerful and complete, they are often more complex and computationally intensive than aggregation-based techniques. Furthermore, their population-based, iterative nature does not lend itself as easily to integration within a gradient-based learning framework like DFL, which typically relies on a differentiable or sub-differentiable path from prediction to decision.

This work seeks to bridge the gap between these two approaches by using a tractable, scalarization-based DFL training loop that is informed by knowledge of the complete, true Pareto Frontier. We avoid the theoretical pitfalls of naïve scalarization by using a Pareto-informed loss function, ParetoSPO loss, which selects a target from the true Pareto Front leveraging a Multi-Objective solver. This allows it to retain the computational efficiency of single-objective DFL while benefiting from the global perspective of Pareto-based methods, representing a new and effective approach to Pareto-informed, decision-focused learning. The unique nature of DFL, which has optimal decisions based on true costs, enables us to get a signal.

3.3 Proposed method: ParetoSPO

Our proposed `ParetoSPO` method, as outlined in Figure 2, is designed for operational contexts where a known, discrete set of user preferences exists. To cater to these, we train a set of models, a model committee, $\{f_{\theta_1}, \ldots, f_{\theta_N}\}$, where each model f_{θ_p} is specialized for a known preference $p \in \{p_1, \ldots, p_N\}$ from the set of the N known preferences. This model takes features x as input to generate a cost prediction $\hat{c}$ that minimizes decision regret specifically for its assigned preference p.

In order to obtain the true Pareto Front for training, we use a Multi-Objective Combinatorial solver to find optimal decisions based on historical costs. Because this solver is slower than a typical single-objective one, we perform this entire calculation offline during data preparation.

During training, we select a single, actionable solution from the Pareto Front using the model's associated preference vector $p \in \mathbb{R}_+^K$. At inference time, the expert model corresponding to a given user preference is used to predict costs.

The core innovation of our method lies in how each expert is trained, the loss function, `ParetoSPOLoss`, leverages knowledge of the true Pareto Front to guide the learning process. To understand its contribution, we first consider the standard approach: A baseline SPO+ model can be extended to handle multiple objectives via linear scalarization. In this approach, the target solution, w^*_{scalar}, would be defined by first scalarizing the *true costs* ($c_{scalar} = c^T p$), then solving the single-objective problem. However, as established, if the true Pareto Front is non-convex, this scalarized solution may be a poor proxy for the user's desired trade-off.

`ParetoSPO` bypasses this limitation. Instead of deriving a target from scalarized costs, we select a target directly from the pre-

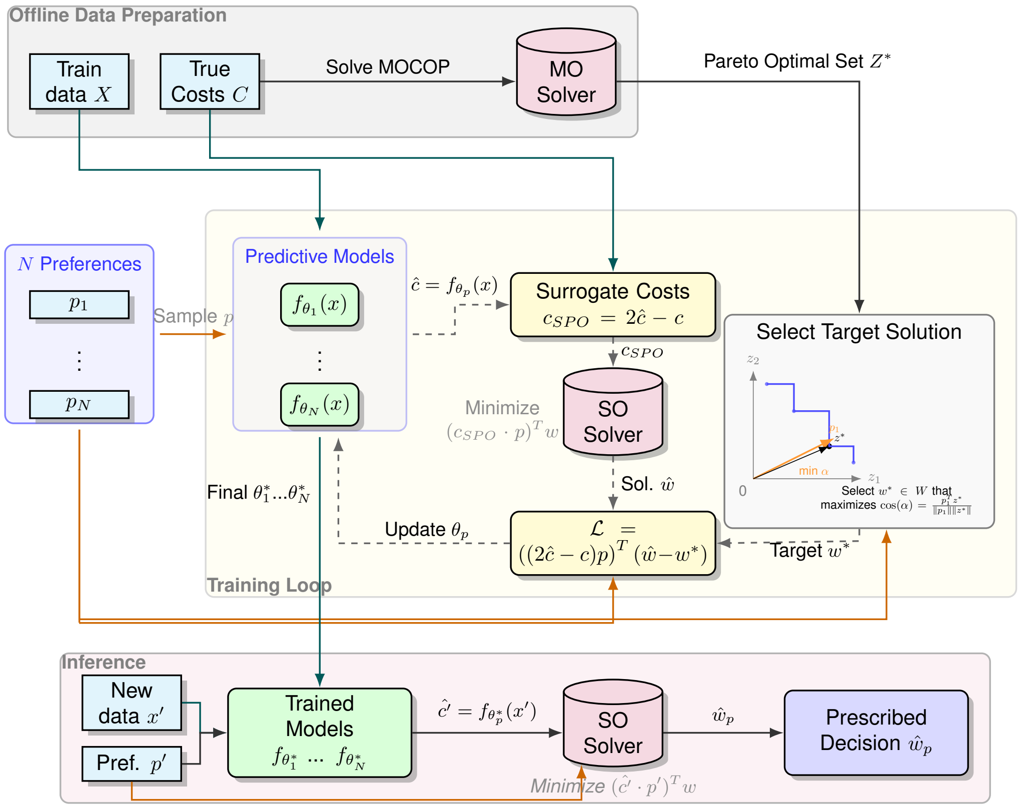

Figure 2. Pareto-informed SPO+ for Multi-Objective Decision Focused Learning

computed true Pareto set W^* and its objective values Z^*. The process, described in detail in Algorithm 1, can be summed up as follows:

1. **Predict Costs:** Given features x and preference p, the expert model f_θ predicts the multi-objective cost vector $\hat{c}$.
2. **Select Target Solution:** To select the "best" solution in the true Pareto set that aligns with the preference vector p, we first normalize the true objective vectors $z^* \in Z^*$ to ensure a fair comparison across objectives. Then, we compute the cosine similarity between the preference vector p and each normalized true objective vector z'^*, obtaining solution $w^* \in W^*$, demonstrating the highest cosine similarity.
3. **Compute Loss:** With the deliberately selected target w^* in hand, we form a surrogate cost vector $c_{SPO} = 2\hat{c} - c$, scalarize it using the preference p, and solve the single-objective problem to get the predicted solution $\hat{w}$. The final loss is: $\mathcal{L} = \left((\hat{w}^T(2\hat{c} - c) - w^{*T}(2\hat{c} - c))p\right)$, shortened to $((2\hat{c} - c)p)^T (\hat{w} - w^*)$.

This hybrid design achieves the best of both worlds: it retains the computational efficiency of using a single-objective solver in the training loop, while transcending the theoretical limitations of simple scalarization. By providing the SPO+ gradient ($2(w^* - \hat{w})$) with a high-quality, Pareto-optimal target w^*, we ensure the learning signal is always directed towards a relevant region of the solution space, even if that region lies on a non-convex part of the Pareto Front. While the model committee architecture means that training time and parameter count scale linearly with the number of preferences, the impact can be mitigated by the fact that the approach is fully parallelizable.

Algorithm 1 Pareto-Informed Decision-Focused Learning

Input: Training data $\{(x_i, c_i)\}_{i=1}^n$, preference set $\mathcal{P}$, learning rate η, number of epochs E
Output: Trained models $f_{\theta_p^*}$

Offline Data Preparation:
for $i = 1$ **to** n **do**
  Compute Pareto Optimal Set: $(Z_i^*, W_i^*) \leftarrow$ MO-Solver(x_i, c_i)

Training Loop:
for *epoch* $= 1$ **to** E **do**
  for $i = 1$ **to** n **do**
    Sample preference $p \sim \mathcal{P}$
    $\hat{c} \leftarrow f_{\theta_p}(x_i)$
    $c_{\text{SPO}} \leftarrow 2\hat{c} - c_i$
    $\hat{w} \leftarrow \arg\min_{w \in \mathcal{W}} (c_{\text{SPO}} p)^T w$
    $z^* \leftarrow \arg\max_{z \in Z^*} \frac{p^\top z}{\|p\| \|z\|}$, $w^* \leftarrow$ solution w $\in W^*$, corresponding to z^*
    $\mathcal{L} \leftarrow ((2\hat{c} - c)p)^T (\hat{w} - w^*)$
    Update θ_p using $\nabla_{\theta_p} \mathcal{L}$

return $f_{\theta_1^*}, \cdots, f_{\theta_N^*}$

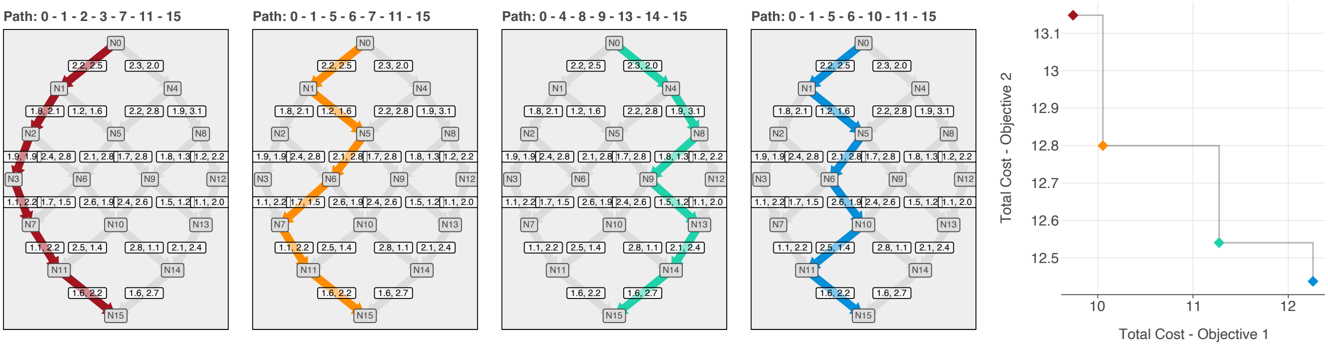

Figure 3. Example - MOSPP solved using a multi-objective solver for a 4x4 grid. Each edge in the network has two costs (c_1, c_2) associated with it. A Multi Objective solver produces a set of Pareto Optimal solutions, each offering a different trade-off of total costs. Path 0-1-2-3-7-11-15 (red) offers the cost that minimizes total cost 1, the first objective, while Path 0-1-5-6-10-11-15 (blue) minimizes the second objective. The other two solutions lie in between.

4 Experimental Setup

To evaluate the performance of our proposed `ParetoSPO` method, we conduct experiments on two classic multi-objective combinatorial optimization problems: the Multi-Objective Shortest Path Problem (MOSPP) and the Multi-Objective Multi-Dimensional Knapsack Problem (MOMDKP). These problems are chosen for their relevance to real-world applications and their well-understood structures.

4.1 Training Configuration

We implement a two-layer feedforward network with 64 hidden units. For both problems, we use a dataset of 2000 instances, split into 70% for training, 10% for validation, and 20% for testing. The models are trained for up to 100 epochs; we select the checkpoint with the best validation performance and report its results on the test set. To solve the underlying multi-objective combinatorial optimization problems, we employ the CP-SAT solver from OR-Tools [5]. For the MOSPP problem, we use a grid size=(5x5) and, for the MOMDKP, we use nitems=32. For the single-objective solver in the shortest path problem, we use a more efficient Bellman-Ford algorithm [4] in place of a generic LP solver.

4.2 Baselines

For both problems, we compare against three baselines:

1. **Average Cost Baseline:** A non-learning baseline that uses the historical mean of objective coefficients to derive decisions. An improvement over this baseline confirms that feature-based learning is effective.
2. **Two-Stage (MSE):** A standard "predict-then-optimize" approach. First, a model is trained to predict the objective vectors by minimizing Mean Squared Error (MSE). Second, these predicted objectives are passed to the solver.
3. **SPO+ (LS):** A decision-focused baseline that applies the standard SPO+ loss to a linearly scalarized version of the multi-objective problem. This represents the most direct adaptation of existing DFL methods to the multi-objective setting.

4.3 Multi-Objective Combinatorial Problems

4.3.1 Multi-Objective Shortest Path Problem (MOSPP)

The MOSPP is a fundamental problem in network optimization. Given a directed graph $G = (V, E)$, the goal is to find a path from a start node $s \in V$ to a terminal node $t \in V$ that simultaneously minimizes a set of k objectives. Each edge $e \in E$ has a cost vector $c_e \in \mathbb{R}^k_+$. A solution is a binary vector $w \in \{0, 1\}^{|E|}$ indicating the selected edges in a path. The problem can be formulated as:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \left(\sum_{e \in E} c_{e,1} w_e, \ldots, \sum_{e \in E} c_{e,k} w_e \right) \\ \text{subject to} \quad & \sum_{e \in \delta^+(v)} w_e - \sum_{e \in \delta^-(v)} w_e = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}, \quad \forall v \in V \\ & w_e \in \{0, 1\}, \quad \forall e \in E \end{aligned} \tag{2}$$

where $\delta^+(v)$ and $\delta^-(v)$ are the sets of outgoing and incoming edges for node v, respectively. For our experiments, we use $k = 2$ objectives on a grid graph. An example of a solved MOSPP problem with known costs is shown in Figure 3, alongside the corresponding Pareto Frontier.

4.3.2 Multi-Objective Multi-Dimensional Knapsack Problem (MOMDKP)

The MOMDKP is a generalization of the classic knapsack problem and serves as a benchmark for resource allocation. Given a set of m items, the goal is to select a subset to pack into a knapsack with q capacity constraints, while maximizing k value objectives. A solution $w \in \{0, 1\}^m$ indicates which items are selected. Formally, the problem is:

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & (V_1^T w, \ldots, V_k^T w) \\ \text{subject to} \quad & C_j^T w \leq K_j, \quad \forall j \in \{1, \ldots, q\} \\ & w_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \end{aligned} \tag{3}$$

where $V_1, \ldots, V_k \in \mathbb{R}^m_+$ are the value vectors, $C_j \in \mathbb{R}^m_+$ are the weight vectors, and K_j are the capacities. To fit this into our cost-minimization method, we treat the values as negative costs, i.e., $c = -V$. We use $k = 2$ objectives and $q = 2$ constraints. An example of

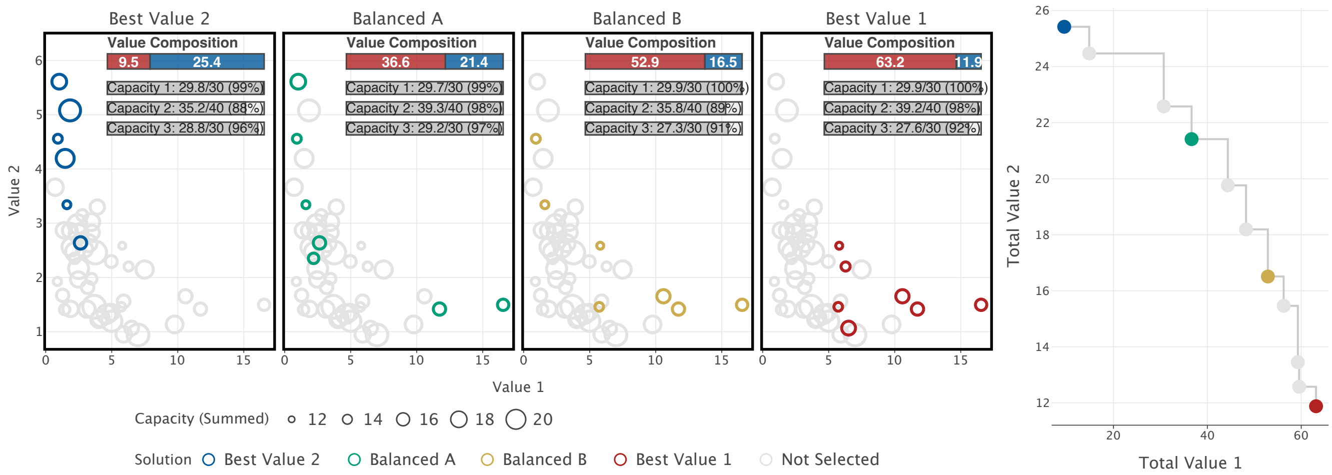

Figure 4. Example - MOMDKP problem solved using a multi-objective solver for fixed coefficients (values). Here, each item has two values associated with it, eg. monetary value, sentimental value, and 3 constraints with capacities of [30, 40, 30], eg. weight, volume, quantity, of the item. A Multi Objective knapsack solver returns a set of Pareto Optimal solutions with different tradeoffs.

a solved MOMDKP problem with known costs is shown in Figure 4, alongside the corresponding Pareto Frontier.

4.4 Data Generation

We use datasets from the PyEPO framework [25] as a foundation for both problems. We extend these datasets by generating two-dimensional coefficients from external features, applying a similar method to produce costs for the MOSPP and values for the MOMDKP.

For n instances, we generate a feature vector $x \in \mathbb{R}^d$ from a standard normal distribution, $x \sim \mathcal{N}(0, I_d)$, where d is the number of feature dimensions. A shared underlying factor, Bx, is computed via a linear transformation, $Bx = \frac{xB^T}{\sqrt{d}} \in \mathbb{R}^\nu$, where $B \in \{0,1\}^{\nu \times d}$ is a fixed binary matrix from a Bernoulli(0.5) distribution, and ν is the number of decision variables ($\nu = |E|$ for MOSPP, $\nu = m$ for MOMDKP). Two conflicting coefficient vectors(cost/value), c_1 and c_2, are then defined as opposing functions of this factor, controlled by a degree parameter, deg:

$$c_1 = \left[\alpha_1 \frac{(Bx + \beta_1)^{\text{deg}}}{\gamma_1^{\text{deg}}} + 1.0 \right] \odot \zeta_1, \tag{4}$$

$$c_2 = \left[\alpha_2 \frac{(\beta_2 - Bx)^{\text{deg}}}{\gamma_2^{\text{deg}}} + 1.0 \right] \odot \zeta_2 \tag{5}$$

The scaling parameters (α, β, γ) are consistent for both the MOSPP and MOMDKP problems, with the sole exception of α_1, which is set to 1 and 5 respectively to maintain consistency with the PyEPO framework. The shared parameter values are ($\alpha_2 = 1, \beta_1 = 3, \beta_2 = 5, \gamma_1 = 3.5, \gamma_2 = 5$). Terms ζ_1, ζ_2 represent multiplicative noise vectors that simulate real-world stochasticity, where each element is drawn independently from a uniform distribution $\zeta \sim U(1-\epsilon, 1+\epsilon)$, with ϵ being the noise width. In the next section, we show our results across different choices of ϵ.

4.5 Performance Metric: Hypervolume Regret

We evaluate the performance of our approach using Hypervolume Regret. The Hypervolume (HV) indicator [2] is a widely used metric for evaluating the quality of a set of Pareto Front points. The hypervolume $hv(Z)$, measures the volume of the objective space dominated by the points in a set Z and bounded by a reference point r. Given the true Pareto Front in the objective space, Z^*, and the set of decisions made with predicted costs, $\hat{Z}$, we define the hypervolume regret R_{hv} as the percentage of hypervolume lost relative to the true hypervolume:

$$R_{hv} = \frac{hv(Z^*) - hv(\hat{Z})}{hv(Z^*)} \tag{6}$$

where $hv(Z) = \text{volume}\left(\bigcup_{z \in Z} \{q \in \mathbb{R}^2 | z \succeq q \succeq r\}\right)$. This normalized metric is highly interpretable: a value of 0 indicates a perfect prediction of the Pareto Front, while a value of 1 indicates that the set of decisions is unable to cover any proportion of the trade-off space. Figure 5 illustrates the concept of the hypervolume regret for a minimization problem. For maximization problems, the objective space is conceptually inverted, and the hypervolume is calculated with respect to a nadir point (e.g., the origin) instead of a utopian reference point.

5 Results

We present the results of our experiments in Table 1. The performance of each method is evaluated based on the mean Hypervolume Regret (R_{hv}) across the test set, under varying levels of noise.

Our results indicate a clear performance hierarchy. The standard MSE baseline performs reasonably well but is consistently outperformed by the decision-focused methods. The scalarized SPO+ baseline shows a marked improvement over MSE, demonstrating the value of incorporating the optimization task into the learning objective.

Notably, our proposed `ParetoSPO` method consistently outperforms the three baselines in both the problem domains and all levels

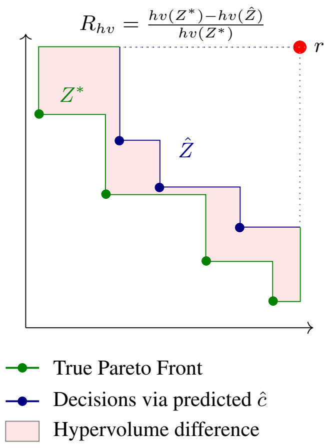

Figure 5. Visual representation of the Hypervolume Regret.

Table 1. Mean Hypervolume Regret (%) for MOSPP and MOMDKP. We report the test data performance with each model finalized based on validation set performance. Lower value is better; a value of 0 implies model recovered the optimal Pareto Frontier.

Problem	Method	ϵ = 0	ϵ = 0.25	ϵ = 0.5
MOSPP	Avg. Cost	13.90 ± 2.69	17.28 ± 3.18	24.57 ± 4.14
	MSE	1.15 ± 0.06	5.51 ± 0.23	13.43 ± 0.56
	SPO+ (LS)	0.98 ± 0.07	5.10 ± 0.26	12.47 ± 0.48
	ParetoSPO	**0.70** ± **0.04**	**4.82** ± **0.20**	**12.00** ± **0.31**
MOMDKP	Avg. Cost	32.87 ± 6.46	37.53 ± 7.76	46.63 ± 8.15
	MSE	12.65 ± 4.60	20.41 ± 6.35	33.48 ± 5.96
	SPO+ (LS)	10.96 ± 4.20	19.27 ± 6.03	31.80 ± 5.96
	ParetoSPO	**8.36** ± **3.32**	**16.67** ± **5.10**	**30.98** ± **5.19**

of noise. Even in the low-noise setting, standard SPO+ baseline, limited by linear scalarization, struggles to find solutions in these non-convex regions. In contrast, `ParetoSPO`, leveraging knowledge of the true Pareto Front to select its target, successfully guides the model towards these complex trade-offs, resulting in a significantly lower hypervolume regret. As noise increases, all methods degrade, but `ParetoSPO` maintains a clear advantage.

6 Discussion, Limitations and Future Work

The experimental results confirm that `ParetoSPO` effectively addresses the research gap identified at the intersection of DFL and multi-objective optimization. Existing methods have forced a choice between the computational tractability of scalarization-based DFL and the theoretical completeness of Pareto-based optimization. Our method provides a practical and effective way to achieve the best of both worlds.

The core innovation is the design of the `ParetoSPOLoss` function. By determining the target solution w^* from the true Pareto Front, rather than from a potentially flawed scalarization of the true costs, we provide a much richer and more geometrically sound learning signal. This allows the model to learn predictions that induce plans in the correct region of the objective space, even if that region is non-convex and thus invisible to the standard SPO+ method. The superior performance of `ParetoSPO` directly quantifies the value of incorporating this explicit knowledge of the Pareto Front's geometry into the decision-focused learning process.

This work advances DFL by providing a practical methodology for building AI systems that can be used for decision-making and planning. It demonstrates how to retain the computational efficiency of single-objective DFL while reaping the theoretical benefits of a Pareto-aware approach.

While our proposed `ParetoSPO` method demonstrates strong empirical performance, it is important to acknowledge its limitations, which in turn motivate several promising directions for future research.

The primary limitation is the scalability of the "model committee" architecture. Our approach requires training a separate, specialized model for each known user preference. This means that training time scales linearly with the number of discrete preferences. While this process is highly parallelizable, it may become computationally prohibitive for problems with a very large or continuous spectrum of user preferences.

Furthermore, this architecture inherently lacks the ability to generalize to new, unseen preferences. Each expert model is trained exclusively on data corresponding to its single, fixed preference vector and has no knowledge of other trade-offs. Consequently, if a user presents a preference that was not part of the original training set, the system cannot generate a tailored, high-quality decision.

These limitations highlight a clear and compelling path for future work. The most significant next step is to overcome the scalability and generalization issues by developing a single, preference-conditioned model. Such a model would take both the feature vector x and the preference vector p as inputs, learning a continuous mapping from preferences to optimal decisions. This could potentially be achieved using techniques from meta-learning or by adapting scalable multi-objective learning architectures like COSMOS [19] to the DFL paradigm. Success in this area would represent a major advance towards building truly flexible and adaptive planning systems.

Additional future work includes applying `ParetoSPO` to more complex, real-world planning problems beyond the benchmarks used here, as well as integrating more sophisticated preference elicitation methods to handle scenarios where user preferences are not explicitly known beforehand.

7 Conclusion

In this paper, we introduced `ParetoSPO`, a novel decision-focused learning framework designed to address the unique challenges of multi-objective combinatorial optimization problems. By integrating a Pareto-informed target selection mechanism into the SPO+ loss function, our method successfully overcomes the established limitations of linear scalarization, leading to superior decision quality, particularly for problems with non-convex Pareto Frontiers.

Our experiments on the Multi-Objective Shortest Path and Multi-Objective Knapsack problems demonstrate that `ParetoSPO` consistently and significantly outperforms both traditional two-stage and standard scalarized DFL baselines across various noise levels. This work provides a practical and effective methodology for building AI systems that can learn to make high-quality, preference-aligned decisions under uncertainty. By showing how to bridge the gap between the computational efficiency of single-objective DFL and the theoretical completeness of Pareto-based methods, we offer a blueprint for a new class of DFL loss functions where the intelligent selection of the learning target is a key design consideration.

References

[1] F. Ahmed and K. Deb. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing*, 17(7):12831299, Dec. 2012. ISSN 1433-7479. doi:

10.1007/s00500-012-0964-8. URL http://dx.doi.org/10.1007/s00500-012-0964-8.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal -distributions and the choice of the reference point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, page 87102, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584140. doi: 10.1145/1527125.1527138. URL https://doi.org/10.1145/1527125.1527138.

[3] C. Bazgan, F. Jamain, and D. Vanderpooten. Discrete representation of the non-dominated set for multi-objective optimization problems using kernels. *European Journal of Operational Research*, 260(3):814827, Aug. 2017. ISSN 0377-2217. doi: 10.1016/j.ejor.2016.11.020. URL http://dx.doi.org/10.1016/j.ejor.2016.11.020.

[4] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):8790, Apr. 1958. ISSN 1552-4485. doi: 10.1090/qam/102435. URL http://dx.doi.org/10.1090/qam/102435.

[5] CPAIOR. Cpaior 2020 master class: Constraint programming, Sept. 2020. URL https://youtu.be/lmy1ddn4cyw.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182197, Apr. 2002. ISSN 1089-778X. doi: 10.1109/4235.996017. URL http://dx.doi.org/10.1109/4235.996017.

[7] A. N. Elmachtoub and P. Grigas. Smart predict, then optimize. *Management Science*, 68(1):926, Jan. 2022. ISSN 1526-5501. doi: 10.1287/mnsc.2020.3922. URL http://dx.doi.org/10.1287/mnsc.2020.3922.

[8] A. N. Elmachtoub, H. Lam, H. Zhang, and Y. Zhao. Estimate-then-optimize versus integrated-estimation-optimization versus sample average approximation: A stochastic dominance perspective, 2023. URL https://arxiv.org/abs/2304.06833.

[9] E. Hauer. Speed and safety. *Transportation Research Record: Journal of the Transportation Research Board*, 2103(1):1017, Jan. 2009. ISSN 2169-4052. doi: 10.3141/2103-02. URL http://dx.doi.org/10.3141/2103-02.

[10] J. Jahn. *Scalarization in Multi Objective Optimization*, page 4588. Springer Vienna, 1985. ISBN 9783709128220. doi: 10.1007/978-3-7091-2822-0_3. URL http://dx.doi.org/10.1007/978-3-7091-2822-0_3.

[11] D. Jia, M. Cao, W. Hu, J. Sun, H. Li, Y. Wang, W. Zhou, T. Yin, and R. Qian. Multi-objective combinatorial optimization algorithm based on asynchronous advantage actorcritic and graph transformer networks. *Electronics*, 13(19):3842, Sept. 2024. ISSN 2079-9292. doi: 10.3390/electronics13193842. URL http://dx.doi.org/10.3390/electronics13193842.

[12] M. Khouadjia, M. Schoenauer, V. Vidal, J. Dreo, and P. Savéant. Pareto-based multiobjective ai planning. In *IJCAI International Joint Conference on Artificial Intelligence*, 08 2013.

[13] S. Kuhlemann, P. Mehta, and V. Tiefenbeck. Greener or Faster: Unraveling the Impact of Travel Time Presentation on Rail vs. Air Travel Decision-Making. In *International Conference on Information Systems (ICIS) 2024*, 2024.

[14] X. Lin, H.-L. Zhen, Z. Li, Q. Zhang, and S. Kwong. Pareto multi-task learning. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS)*, pages 12037–12047, 2019.

[15] M. B. Mandell. Modelling effectiveness-equity trade-offs in public service delivery systems. *Management Science*, 37(4):467482, Apr. 1991. ISSN 1526-5501. doi: 10.1287/mnsc.37.4.467. URL http://dx.doi.org/10.1287/mnsc.37.4.467.

[16] J. Mandi, E. Demirovic, P. J. Stuckey, and T. Guns. Smart predict-and-optimize for hard combinatorial optimization problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):16031610, Apr. 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i02.5521. URL http://dx.doi.org/10.1609/aaai.v34i02.5521.

[17] J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:16231701, Aug. 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15320. URL http://dx.doi.org/10.1613/jair.1.15320.

[18] V. V. Mii and G. Perakis. Data analytics in operations management: A review. *Manufacturing and Service Operations Management*, 22(1):158169, Jan. 2020. ISSN 1526-5498. doi: 10.1287/msom.2019.0805. URL http://dx.doi.org/10.1287/msom.2019.0805.

[19] M. Ruchte and J. Grabocka. Scalable pareto front approximation for deep multi-objective learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1306–1311, 2021. doi: 10.1109/ICDM51629.2021.00162.

[20] M. Ruchte and J. Grabocka. Multi-task problems are not multi-objective, 2021. URL https://arxiv.org/abs/2110.07301.

[21] M. RuSS and G. Gust. Reliable route planning and time savings in real-world urban intermodal transportation networks: Evidence from hamburg, germany. *Expert Systems with Applications*, 227:120196, Oct. 2023. ISSN 0957-4174. doi: 10.1016/j.eswa.2023.120196. URL http://dx.doi.org/10.1016/j.eswa.2023.120196.

[22] N. Schutte, K. Postek, and N. Yorke-Smith. Robust losses for decision-focused learning. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, IJCAI-2024, page 48684875. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024. doi: 10.24963/ijcai.2024/538. URL http://dx.doi.org/10.24963/ijcai.2024/538.

[23] S. Shah, B. Wilder, A. Perrault, and M. Tambe. Leaving the nest: Going beyond local loss functions for predict-then-optimize. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):1490214909, Mar. 2024. ISSN 2159-5399. doi: 10.1609/aaai.v38i13.29410. URL http://dx.doi.org/10.1609/aaai.v38i13.29410.

[24] B. Tang and E. B. Khalil. *Multi-task Predict-then-Optimize*, page 506522. Springer International Publishing, 2023. ISBN 9783031445057. doi: 10.1007/978-3-031-44505-7_34. URL http://dx.doi.org/10.1007/978-3-031-44505-7_34.

[25] B. Tang and E. B. Khalil. Pyepo: a pytorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, July 2024. ISSN 1867-2957. doi: 10.1007/s12532-024-00255-x.

[26] T. Vanderschueren, T. Verdonck, B. Baesens, and W. Verbeke. Predict-then-optimize or predict-and-optimize? an empirical evaluation of cost-sensitive learning strategies. *Information Sciences*, 594:400415, May 2022. ISSN 0020-0255. doi: 10.1016/j.ins.2022.02.021. URL http://dx.doi.org/10.1016/j.ins.2022.02.021.

[27] M. Voorneveld. Characterization of pareto dominance. *Operations Research Letters*, 31(1):711, Jan. 2003. ISSN 0167-6377. doi: 10.1016/s0167-6377(02)00189-x. URL http://dx.doi.org/10.1016/S0167-6377(02)00189-X.