One-Size-Fits-All: Robust Solutions to Multi-Objective Planning Problems

Nils Wilde Dylan Shell

***Abstract*— Real-world robot planning problems need to consider multiple competing objectives. To obtain appropriate trade-offs, it is common for a single-objective problem to be posed via some parametrized scalarization. But the parameters, representing the relative importance or bounds of the individual objectives, are often subjective. Tuning them requires careful calibration to end-user needs and adaptation to specific use cases. Imperfect parameter values are likely to be identified even if users are queried directly as what can be adduced from opinions is inherently imprecise, or may reflect the preference of but one user among many. Thus, we study *robust solutions*: a single, one-size-fits-all solution for linear scalarization of objectives that minimizes the worst-case error. Building on recent work on Pareto-exploration, we present a pair of search algorithms that produce such solutions. We report results based on 4 different problems, evaluating performance versus sample budget. Further, we apply these methods to address the problem of finding a single, fair preference suitable for a set of users.**

I. Motivation and Introduction

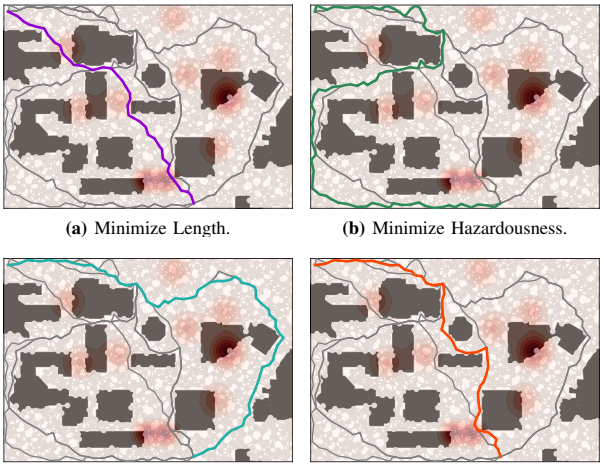
The guilty secret that we roboticists keep to ourselves is that deploying any robot involves laborious tuning of many parameters, hyper-parameters, and empirically determined constants. This is true regardless of whether a classical or more modern data-driven approach is used. We attack this concern head-on in the context of robot *motion planning*.

A great deal of parameter tuning deals with relative prioritization of competing desiderata. A typical case involves an objective/reward function with a summation of weighting terms that encode aspirations [1]. Figure 1 is an illustrative example of the problem of selecting weightings for motion planning objectives, balancing the twin aims of minimizing distance and reducing the hazards encountered.

Such problems are tidily handled via the theory of multi-objective optimization,[1] which employs the Pareto front (or frontier) as a way of summarizing candidates; the frontier set may be conceived of as a visualization that enables an informed decision to be made by looking, not at a single potential solution, but at a solution in comparison to other possibilities. We ultimately require that a single solution be selected because the robot must execute *some* path. Though that final choice corresponds to the solution of a standard single objective optimization problem, precisely *which* optimization problem will not be known until enough solutions under multiple objectives have been produced.

The authors are with Departments of Computer Science, respectively, at Dalhousie University, Halifax, NS, Canada and Texas A&M University, College Station, TX, USA. nils.wilde@dal.ca & dshell@tamu.edu.

[1]Adoption of multi-objective methods in the robotics community still represents a small minority (see [2]). This was itself a topic of discussion at a recent workshop [3]. There are myriad reasons why this is so, but two practical ones are computational expense and preference indefiniteness.



(a) Minimize Length. **(b)** Minimize Hazardousness. **(c)** Equal importance weights. **(d)** One-size-fits-all solution.

Fig. 1: A vehicle plans to traverse the environment given a map indicating obstacles (dark regions) and hazards (Gaussians giving a probability of encountering a safety violations, shown in red). Distinct paths differ: some traversals minimize solely distance (a), others are less hazardous but incur increases in distance (b), or are a compromise of the two (c) & (d). A uniform weighting gives (c), but the solution we propose produces (d).

This entails two crucial ideas: (i) multi-objective optimization delays the prioritization of objectives, which can be interpreted as expressing uncertainty; (ii) that uncertainty is resolved by making a choice informed in relation to other potentialities, i.e., it is not strictly local. While these seem obvious, the former is seldom explicit in typical presentations of the notion of Pareto optimality in robotics, while the latter is lost once either a scalarization is given or assumed.

Producing the Pareto front is not the full solution: a single selection has to be obtained for the robot to actually execute. Thus, input is usually solicited from a person, who makes an *a posteriori* assessment of the Pareto front, and selects a point in order to express their preference. However, that point may fail to provide unambiguous preference information: a single point —especially if a corner or elbow of the curve— will usually correspond to a set of weightings. Further, that under-specified information can then fail to generalize to other instances. And there's no reason to suppose it would be representative of a broad class of users. Together, these form the problem we term 'preference indefiniteness'.

This paper offers a formulation based on explicit risk-minimization to mitigate preference indefiniteness. We present algorithms that, based on few samples, can obtain high-quality results without constructing an entire Pareto front. We build directly on the recent work [4], a method that *does* seek to approximate the entire Pareto front. In addition to an improved definition of regret, we establish a geometric optimality criterion that leads to informed algorithms which

consistently demonstrate superior performance in terms of optimization evaluations when solving the proposed problem.

Related work and Context: The idea that many motion planning problems involve trade-offs between different cost functions and that an appropriate solution concept is that of Pareto optimality is well known (e.g., see textbook [5, §7.7.2; §9.1.1.1]). For discretized problems, the topic has seen considerable work from the heuristic search community, with a summary of progress (as of 2023) in [6]. For some problems —e.g., clearance from planar obstacles *vs* path length— a parameterized Pareto front can be computed directly as a solution, but this is rare and methods achieving such are therefore notable [7], [8]. More often, points along the Pareto front are sampled, and algorithms (such as Adapted Weighted Sum [9], Min-Regret Pareto Sampling [4], or RL-based methods [10]) have been developed to give approximations of high quality.

There is a pervasive view that if only one knew the 'missing' priorities/weights/parameters, the problem would be simple, and require standard scalar treatment. A closely related (in fact we believe overlapping) problem of reward learning in HRI [11], [12] considers user preferences on exhibited behavior and can be formulated as finding weightings that prioritize between competing sub-objectives. Most active learning techniques avoid construction of the Pareto front, focusing instead on various means to solicit effective user input [13], [14]. (An exception is [15] which first constructs the front.) Point (ii) above emphasizes, as recognized in the classical decision analysis literature [16] and in the reward learning literature now, that expressing preferences across hypothetical options is much more difficult than selecting among concrete choices. As mentioned previously, our perspective is that (ii) illustrates a slightly different fact: a scalarization choice is less about prescribing the absolute position of a point than about expressing its relation compared to a neighbourhood of options.

Contributions: *1)* A novel framing of multi-objective problems, focusing on identifying a single solution, based on consideration of nearby solutions to preserve a notion of uncertainty (Sec. I and II). *2)* Theoretical foundations for the problem including an improved definition of regret and the introduction of a geometric optimality criterion (Sec. III-B). *3)* Two algorithms, one based on gradient descent (Sec. III-C) and the other adapting the Pareto-sampling method from [4] (Sec. III-D). *4)* Experimental results with 2- and 3-dimensional Pareto fronts (Sec. IV) showing sampling efficiency of our methods and their value in selecting compromise weightings for preferences among multiple users.

II. Problem Statement

In multi-objective optimization problems (MOOP), the key notion of Pareto optimality respects the fact that problems involving multiple criteria yield solutions that are only partially ordered in the absence of other information. The solution concept is that of a set where each element could be the top choice under some prioritization of criteria. For $n \in \mathbb{N}$, a general MOOP is of the form

$$\min_{s \in \mathcal{S}} \Big(f_1(s), f_2(s), \ldots, f_n(s) \Big). \tag{1}$$

Here, the set of feasible solutions given constraints is denoted $\mathcal{S}$, and it is desired to simultaneously minimize n objectives $f_i(s), i = 1, \ldots, n$.

Commonly, robot planning and control tackles MOOPs by means of scalarization to combine the different objectives into a single scalar objective function [17]. Popular approaches include linear scalarization, *i.e.,* a weighted summation of objectives [4], and weighted maximization [18]. We introduce the notion of a parametrized scalarization:

Definition 1 (Parametrized scalarization)**.** Given objective functions $f_1(s), f_2(s), \ldots, f_n(s)$ a *parametrized scalarization* is a function $h : \mathcal{S} \to \mathbb{R}$ with m tunable parameters $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]$, such that for fixed parameters $\boldsymbol{w}$ and two solutions s and s' where $f_i(s) \leq f_i(s')$ for all $i = 1, \ldots, n$, we have $h(s, \boldsymbol{w}) \leq h(s', \boldsymbol{w})$.

For convenience of presentation, we assume that these parameters are chosen from the set $\mathcal{W} = [0, 1]^m$. The choice of parameters $\boldsymbol{w}$ allows for attaining different solutions to the MOOP by solving the single-objective problem

$$u(\boldsymbol{w}) = \min_{s \in \mathcal{S}} h(s, \boldsymbol{w}). \tag{2}$$

Yet, the *correct* parameters are often subjective since they depend on decision maker's —*e.g.,* end-user's— preference for the attained solution s. We define the set of optimal solutions to (2) for fixed parameters $\boldsymbol{w}$ as

$$\sigma(\boldsymbol{w}) = \Big\{ s \in \mathcal{S} \;\Big|\; h(s, \boldsymbol{w}) = u(\boldsymbol{w}) \Big\}, \quad \forall \boldsymbol{w} \in \mathcal{W}.$$

This is a sort of set-valued '$\arg\min$' because uniqueness cannot be ensured (in general, we know neither that f_is are injective, nor whether h will always 'enable' every f_i).

In this work, we propose a method to compute a weight vector $\boldsymbol{w}' \in \mathcal{W}$ that will, for any $\bar{\boldsymbol{w}} \in \mathcal{W}$, produce solutions $\sigma(\boldsymbol{w}')$ that would not be too objectionable to someone interested in solution $s \in \sigma(\bar{\boldsymbol{w}})$. Building on the notion of *regret* in [4] for linear scalarization, we generalize definition:

Definition 2 (Regret)**.** Given two weights $\boldsymbol{w}', \bar{\boldsymbol{w}} \in \mathcal{W}$, the regret of $\boldsymbol{w}'$ under $\bar{\boldsymbol{w}}$ is defined as

$$\begin{aligned} r(\boldsymbol{w}' | \bar{\boldsymbol{w}}) &= \max_{s' \in \sigma(\boldsymbol{w}')} h(s', \bar{\boldsymbol{w}}) - h(s(\bar{\boldsymbol{w}}), \bar{\boldsymbol{w}}) \\ &= \max_{s' \in \sigma(\boldsymbol{w}')} h(s', \bar{\boldsymbol{w}}) - u(\bar{\boldsymbol{w}}). \end{aligned} \tag{3}$$

Intuitively, $r(\boldsymbol{w}' | \bar{\boldsymbol{w}})$ represents an error in cost incurred by using an optimal solution given parameters $\boldsymbol{w}'$ to approximate a solution given parameters $\bar{\boldsymbol{w}}$. The $\max$ with s' over $\sigma(\boldsymbol{w}')$ is because, having obtained some solution with $\boldsymbol{w}'$, we conservatively assume that is the worst-case element of S that produces $u(\boldsymbol{w}')$. (Notice that the ambiguity is irrelevant because, for all $s_1, s_2 \in \sigma(\bar{\boldsymbol{w}})$, $h(s_1, \bar{\boldsymbol{w}}) = h(s_2, \bar{\boldsymbol{w}})$.)

Simplification 1. There are circumstances wherein (3) can be simplified considerably, eliding the '$\max$' over solutions.

This is when $\arg\max$ solutions tied under one weighting will not differ under the weighting they are approximating.

$$\text{If } s_1 \in \sigma(\boldsymbol{w}') \ \wedge \ s_2 \in \sigma(\boldsymbol{w}') \implies \big[h(s_1, \bar{\boldsymbol{w}}) = h(s_2, \bar{\boldsymbol{w}})\big],$$
$$\begin{aligned} \text{then} \quad r(\boldsymbol{w}'|\bar{\boldsymbol{w}}) &= h(s(\boldsymbol{w}'), \bar{\boldsymbol{w}}) - h(s(\bar{\boldsymbol{w}}), \bar{\boldsymbol{w}}) \\ &= h(s(\boldsymbol{w}'), \bar{\boldsymbol{w}}) - u(\bar{\boldsymbol{w}}), \qquad (4) \\ &\text{where } s(\boldsymbol{w}') = h(s_0, \bar{\boldsymbol{w}}) \text{ for any } s_0 \in \sigma(\boldsymbol{w}'). \end{aligned}$$

Observe that when $h(\cdot, \boldsymbol{w})$ is convex for every $\boldsymbol{w} \in \mathcal{W}$, one may employ (4) in place of (3), as $|\sigma(\boldsymbol{w})| = 1$.

We adopt the form in Eq. (4) as the optimizers we will employ in practice provide only a single solution—this obviates the separate 'max' over solutions. So long as the optimizers yield consistent output (i.e., behave deterministically), then the notion of risk under Simplification 1 is well defined.

The following is the computational problem we address.

Problem 1 (One-Size-Fits-All (OSFA))**.** Given an MOOP with objective functions $f_1(s), \ldots, f_n(s)$ and a parametrized scalarization $h(s, \boldsymbol{w})$, find the *robust weight* $\boldsymbol{w}'$ that solves

$$\min_{\boldsymbol{w}' \in \mathcal{W}} \max_{\bar{\boldsymbol{w}} \in \mathcal{W}} r(\boldsymbol{w}'|\bar{\boldsymbol{w}}). \qquad (5)$$

The most common scalarization technique in robot planning is linear scalarization, *i.e.,* a weighted sum [4], of form

$$h(s, \boldsymbol{w}) = \sum_{i=1}^{n} w_i \cdot f_i(s) = \boldsymbol{w} \cdot \boldsymbol{f}(s). \qquad (6)$$

Simplification 2. (Linear Scalarization) In the remainder of the paper, we restrict ourselves to consideration of $h(s, \boldsymbol{w})$ functions of the form in Eq. (6). We follow the standard convention of taking $m = n$ and requiring $\sum_{i=1}^{n} w_i = 1$, which with respect to (6) is without any loss of generality.

We note that this case corresponds to the overwhelming majority of scalarization functions in the literature (see §1.1 in [19] for references describing specific use in robotics).

III. Approach

A. Preliminaries: known prior results

The authors of [4] analyze linear scalarization, which we briefly review here. This requires two assumptions:

Bounded Objectives: For any weight $\boldsymbol{w} \in \mathbb{R}^n_{\geq 0}$, and any solution $s(\boldsymbol{w})$, the objectives $\boldsymbol{f}(s(\boldsymbol{w}))$ are bounded.

Exact Solution: An exact solver exists for (2).

The first assumption is mild and can usually be satisfied in practice by appropriate design of cost functions. Using the assumptions, the authors of [4] provide the following three preliminary results: *i*) The optimal cost function $u(\boldsymbol{w})$ is a concave function of $\boldsymbol{w}$ and continuous over $\mathcal{W}$. *ii*) $\boldsymbol{f}(s(\boldsymbol{w}))$ is a sub-gradient, $\partial u(\boldsymbol{w})$, of $u(\boldsymbol{w})$ at $\boldsymbol{w}$. *iii*) For any fixed weight $\boldsymbol{w}'$, the regret $r(\boldsymbol{w}'|\bar{\boldsymbol{w}})$ is a convex function of $\bar{\boldsymbol{w}}$.

B. Optimality condition

Building on these insights, we now establish a sufficient condition for an optimal solution of Problem 1. We define $\boldsymbol{b}^i$ as the cardinal vector along the i-th dimension, and let B be the set of all cardinal vectors, i.e., the $B = \{\boldsymbol{b}^1, \ldots, \boldsymbol{b}^m\}$.

Lemma 1 (Maximum regret weight)**.** Consider some weight $\boldsymbol{w}$. The maximum regret $r(\boldsymbol{w}|\bar{\boldsymbol{w}})$ is attained at one of the basis weights $\boldsymbol{b}^i$, i.e., $\bar{\boldsymbol{w}} = \boldsymbol{b}^i$.

Proof. Bauer's Maximum Principle [20] asserts the value is attained at an extreme point because regret $r(\boldsymbol{w}'|\bar{\boldsymbol{w}})$ is a convex function of $\bar{\boldsymbol{w}}$, and the weights $\bar{\boldsymbol{w}}$ lie in a compact, convex set. The extreme points of $\mathcal{W}$ are just B. □

The lemma implies that to minimize the maximum regret one needs to simultaneously minimize the regret with respect to all basis weights in $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^n$. The linear interpolation of the points $u(\boldsymbol{b}^1), \ldots, u(\boldsymbol{b}^n)$ defines a plane L in $\mathbb{R}^{n+1}$. Equal regret at all basis weights would be achieved by a solution s' with features $\boldsymbol{f}'$ such that $\boldsymbol{f}'$ is parallel to the interpolation plane L. However, such a solution might either not be attainable, e.g., when $u(\boldsymbol{w})$ is piece-wise linear, or not be Pareto-optimal, i.e., s' is not a solution to (2) for any $\boldsymbol{w}' \in \mathcal{W}$. Recalling that $\boldsymbol{f}(s(\boldsymbol{w}))$ is a sub-gradient of $u(\boldsymbol{w})$, the next lemma establishes a key characteristic.

Lemma 2 (Gradient of optimal solution)**.** Supposing Simplification 1 is applicable, an optimal solution to Problem 1 is achieved by a weight $\boldsymbol{w}'$ with features $\boldsymbol{f}' = \boldsymbol{f}(s(\boldsymbol{w}'))$ where $||\boldsymbol{f}' - L||_\infty$ is minimal.

Proof. Let $\boldsymbol{w}^{(0)}$ and $\boldsymbol{w}^{(1)}$ with $\boldsymbol{f}^{(i)} = \boldsymbol{f}(s(\boldsymbol{w}^{(i)}))$ $i \in \{0, 1\}$. Suppose $||\boldsymbol{f}^{(0)} - L||_\infty < ||\boldsymbol{f}^{(1)} - L||_\infty$; identify dimension $\bar{k} \in \{1, \ldots, n\}$ where $|\boldsymbol{f}^{(0)}_j - L_j| < |\boldsymbol{f}^{(1)}_{\bar{k}} - L_{\bar{k}}|$, $\forall j \in \{1, \ldots, n\}$. Owing to $h(s, \boldsymbol{w})$ being linear, then $r(\boldsymbol{w}^{(0)}|\boldsymbol{b}^j) < r(\boldsymbol{w}^{(1)}|\boldsymbol{b}^{\bar{k}}), \forall j \in \{1, \ldots, n\}$. Lemma 1 implies $\max_{\bar{\boldsymbol{w}} \in \mathcal{W}} r(\boldsymbol{w}^{(0)}|\bar{\boldsymbol{w}}) < r(\boldsymbol{w}^{(1)}|\boldsymbol{b}^{\bar{k}})$. The ∞-norm condition, thus, implies $\max_{\bar{\boldsymbol{w}} \in \mathcal{W}} r(\boldsymbol{w}^{(0)}|\bar{\boldsymbol{w}}) < \max_{\bar{\boldsymbol{w}} \in \mathcal{W}} r(\boldsymbol{w}^{(1)}|\bar{\boldsymbol{w}})$, and the solution's existence follows from the lower bound of any norm. □

We illustrate the lemmas in Figure 2 for two objectives; since $w_2 = 1 - w_1$, we plot the function $u(\boldsymbol{w})$ over w_1. The regret equals the first-order approximation error between $\boldsymbol{w}'$ and $\bar{\boldsymbol{w}}$, shown in (a). Vectors $\boldsymbol{b}^1$ and $\boldsymbol{b}^2$ characterize the end points of $u(w)$ and define the interpolation plane L (b). The optimal solution $\boldsymbol{w}'$ has a tangent plane $f(s(\boldsymbol{w}'))$ parallel to L and thus attains equal regret at $r(\boldsymbol{w}'|\boldsymbol{b}^1) = r(\boldsymbol{w}'|\boldsymbol{b}^2)$.

Next, we leverage this insight to develop solution methods.

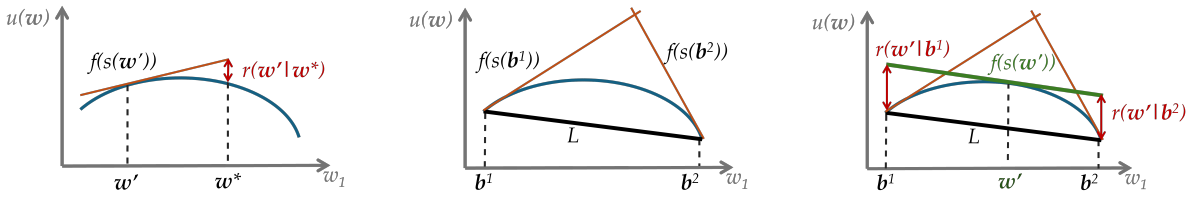
C. Gradient Descent Algorithm

First, we propose a gradient descent–based algorithm to compute $\boldsymbol{w}'$. Lemma 2 provides an error function given an interpolation plane L:

$$g_L(\boldsymbol{w}) = ||\boldsymbol{f}(s(\boldsymbol{w})) - L||_\infty. \qquad (7)$$

However, the partial derivative of the infinity norm is zero for all elements that do not attain the current maximum and thus is not well suited for gradient descent. Therefore, we approximate it using a p-norm. In optimization, the p-root can be dropped and only the terms in the summation are raised to a power p [**?**], resulting in a surrogate error

$$g^p_L(\boldsymbol{w}) = \sum_{i=1}^{n} (f_i(s(\boldsymbol{w})) - L_i)^p. \qquad (8)$$



(a) Regret as the error of a first order approximation of $u(\boldsymbol{w})$.

(b) Basis solution and interpolation plane L.

(c) Optimal solution $\boldsymbol{w}'$ to Problem 1.

Fig. 2: Illustration of regret (a), the basis plane L (b), and an optimal solution to Problem 1 (c). The example has two objectives, such that the weightspace $\mathcal{W}$ is projected to the x-axis (where $w_2 = 1 - w_1$). Blue is the optimal cost $w(\boldsymbol{w})$, orange tangents given by $f(s(\boldsymbol{w}))$, red is the regret $r(\boldsymbol{w}'|\boldsymbol{w})$, black is the interpolation plane L, and green is the tangent of the optimal solution.

Algorithm 1: SIM. OPT. FOR ALL WEIGHTS I

Input: An exact solver to find $s(\boldsymbol{w})$, budget K.
Output: Robust weight $\boldsymbol{w}'$ and maximum regret.
1 Compute basis solutions $s(\boldsymbol{b}^1), \ldots, s(\boldsymbol{b}^n)$ with exact solver
2 Derive interpolation plane L
3 Find $\boldsymbol{w}' \in \arg\min_{\boldsymbol{w}} g^p_L(\boldsymbol{w})$ using gradient descent for $(K - n)$ iterations, starting at $\boldsymbol{w}^0 = [1/n, \ldots, 1/n]$.
4 **return** $\boldsymbol{w}'$

Algorithm 1 provides a concise description of a standard gradient-descent. To construct the error function, we first compute the basis solutions for weights $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^n$ (line 1), and find the interpolation plane L (line 2). We then employ a gradient search over $K - n$ iterations and return the final solution. We refer to this first algorithm as *Simultaneous Optimization for All Weights 1* (SOFA-I).

D. Informed Sampling Algorithm

While SOFA-I provides a heuristic solution to Problem 1, its performance can suffer from the non-linear mapping between weights $\boldsymbol{w}$ and cost vector $\boldsymbol{f}$. As characterized in [4], [21], different weights can result in similar (or even identical) solutions, such that the gradient of (8) changes only slowly (or remains constant) — making gradient descent inefficient. The challenge of weight indefiniteness is explicitly addressed in the regret-based sampling method of Pareto-fronts presented in [4]. Building on their result, we propose a second approach to Problem 1 using an informed sampling algorithm. Our method is an iterative search, starting with the basis weights $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ and incrementally placing new sample weights $\boldsymbol{w}$ until an optimal solution is found, or the maximum number of calls to optimal solver for (2) is exhausted. (The n in $K-n$ is because the budget covers calls in line 1 too. Specialized solutions for the single-objective basis problems may have evaluation costs that differ.)

We briefly revisit the Minimum-Regret Pareto Sampling (MRPS) algorithm from [4], and then present our key modification. A neighbourhood N is defined as a collection of n linearly independent weights, and $\mathcal{C}(N)$ as its convex hull. Given $N = \{\boldsymbol{w}^1, \ldots, \boldsymbol{w}^n\}$, let weight $\tilde{\boldsymbol{w}}(N)$ be the weight within $\mathcal{C}(N)$ that is least approximated by any of the weights $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^n$. Using the notion of regret, this becomes

$$\tilde{\boldsymbol{w}}(N) = \max_{\boldsymbol{w}' \in \mathcal{C}(N)} \min_{\boldsymbol{w} \in N} r(\boldsymbol{w}'|\boldsymbol{w}). \tag{9}$$

While this cannot be efficiently computed, a key result of [4] is that a tight approximation $\bar{\boldsymbol{w}}$ can be found using a linear program (LP). Solving the scalarized problem for the weight $\bar{\boldsymbol{w}}$ results, if found, in a solution different than those corresponding to the neighbourhood weights $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$. MRPS repeatedly i) selects the neighbourhood with maximum regret bounds, ii) computes the solution for $\bar{\boldsymbol{w}}$, and iii) splits the neighbourhood into n smaller neighbourhoods.

Yet, Problem 1 does not seek to approximate the entire Pareto front, but rather seeks a single sample that satisfies Lemma 2, *i.e.,* minimizes (7). Therefore, we modify the algorithm by guiding the neighbourhood selection. Given a neighbourhood, let $\boldsymbol{f}^1, \ldots, \boldsymbol{f}^n$ be the cost vectors for the neighbourhood weights $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^n$. We define $\boldsymbol{\mu}(N)$ as the mean of cost vectors $\boldsymbol{f}^1, \ldots, \boldsymbol{f}^n$. This serves as a heuristic for the error in (7) for any weight in $\mathcal{C}(N)$. Thus, our method selects the neighbourhood N that minimizes $||\mu(N) - L||_\infty$.

Algorithm 2 outlines our informed sampling approach—dubbed *Simultaneous Optimization For All weights 2 (*SOFA-II*)*. To summarize its operation: just like SOFA-I, we begin by computing the basis solutions and finding the interpolation plane L (line 2). We then initialize the set of samples Ω, and a set of neighbourhoods, containing the basis (line 3). Successively, we then select neighbourhood N with the best heuristic value for (7), and compute the min-regret bound weight $\bar{\boldsymbol{w}}(N)$ using the LP from [4] (lines 5–6). We then compute the solution for $\bar{\boldsymbol{w}}(N)$ using the exact solver, store the new sample in the set Ω, and split the neighbourhood N into n new neighbourhoods by substituting $\bar{\boldsymbol{w}}(N)$ (lines 7–13). Finally, we return the weight $\boldsymbol{w}'$ that attains the best objective value among all sampled solutions. Algorithm 2 may also be understood as a specialization of the approach in [4] to return a single solution, with the key guiding idea being the use of an optimality criterion relating $\mu(N)$ to L.

E. Adaptation for prior weight spaces

Problem 1 seeks to minimize the maximum regret over the entire weight set $\mathcal{W}$. As established in Lemma 1, this results in considering the regret incurred at extremal weights where only a single objective is considered. In practical instances, this may be overly conservative since it is unlikely that a user preference —even in the worst case— would completely disregard all but one objective function.

Thus, we introduce a modified version of Problem 1 where prior information is given in form of a set P of linearly independent weights $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^n \in \mathcal{W}$. Such a set may be obtained from sampled or learned user preferences; if more

Algorithm 2: SIM. OPT FOR ALL WEIGHTS II

Input: An exact solver to find $s(\boldsymbol{w})$, budget K
Output: Robust weight $\boldsymbol{w}'$ and maximum regret

Compute basis solutions $s(\boldsymbol{b}^1), \dots, s(\boldsymbol{b}^n)$ with exact solver
Derive interpolation plane L
$\Omega \leftarrow \{\boldsymbol{b}^i | i = 1, \dots, n\}$; $\mathcal{N} \leftarrow \{\Omega\}$
for $k = n$ *to* K **do**
  Select N from $\mathcal{N}$ to min $||\boldsymbol{\mu}(N) - L||_\infty$
  Compute min-regret bound weight $\bar{\boldsymbol{w}}(N)$
  Obtain $s(\bar{\boldsymbol{w}}(N)), \boldsymbol{f}(s(\bar{\boldsymbol{w}}(N)))$ from exact solver
  $\Omega \leftarrow \Omega \cup \{\tilde{\boldsymbol{w}}(N)\}$
  $\mathcal{N} = \mathcal{N} \setminus N$ // Remove expanded neighbourhood
  for $\boldsymbol{w}^i$ *in* N **do**
    $N^i \leftarrow N \setminus \{\boldsymbol{w}^i\} \cup \{\bar{\boldsymbol{w}}(N)\}$ // Substitute weight
    if N^i *is a lin. independent neighbourhood* **then**
      $\mathcal{N} = \mathcal{N} \cup N^i$
Select $\boldsymbol{w}' \in \arg\min_{\boldsymbol{w} \in \Omega} g(\boldsymbol{w})$
return $\boldsymbol{w}'$

than n weights are given, P will be defined as the vertices of their convex hull. Using this prior, the objective is:

$$\min_{\boldsymbol{w}' \in P} \max_{\bar{\boldsymbol{w}} \in \mathcal{W}} r(\boldsymbol{w}' | \bar{\boldsymbol{w}}). \tag{10}$$

We observe that the results from Lemma 1 and 2 directly extend to this case. Further, SOFA-I and SOFA-II can be adapted with minor modification: Taking P as an input, we can compute the basis solutions for the weights in P and define the interpolation with respect to their solutions; the main procedure of both algorithms remains unchanged. This allows for tackling a wider range of practically relevant problems, as will be highlighted in one of our experiments.

IV. RESULTS

We showcase the two methods and the practical benefits of addressing Problem 1 in a set of simulation experiments using different motion planning problems with $n \in \{2, 3\}$.

A. Experiment setup

a) Planning Problems: We consider the following motion planning problems:

i) **DUBINS2D** (adapted from [4]): The problem considers a Dubins vehicle in obstacle-free space with two objectives: trajectory length and discomfort, measured by $1/r^2$ where r is the turning radius. An example is illustrated in Figure 3a.

ii) **DUBINS3D** (adapted from [4]): This is an augmented version of Dubins 2D with integral curvature as third objective.

iii) **DISK SWAP** (inspired by [22]): Two disk robots exchange positions. One robot is the leader and announces its path via an intermediate position; the other robot then optimizes its path accordingly. The two objectives are the path lengths of the both robots, illustrated in Figure 3b.

iv) **HAZARD NAVIGATION**: This solves bi-objective path planning on a graph using A^*. On top of an obstacle environment a map of hazards describes unsafe areas, shown in Figure 3c, and earlier in Figure 1. The objectives are path length and cumulative probability of encountering a hazard.

All planning problems are designed to generate a variety of randomized instances. DUBINS2D and DUBINS3D randomly sample goal configurations, DISK SWAP randomizes the initial distance between the disk robots and HAZARD NAVIGATION randomizes the map of hazards.

b) Experiments: We consider two experiment setups. In Experiment 1 we directly solve Problem 1, *i.e.,* seek to find a one-size-fits-all solution across the entire Pareto front. In Experiment 2 a collection of 10 different user preferences in the form of a set of weights $U = \{\boldsymbol{w}^1, \dots, \boldsymbol{w}^{10}\}$ is provided *a priori*. While we generate these weights using uniformly random sampling over $\mathcal{W}$, such an input may be provided in real-world scenarios from user experience surveys, or a result of preference learning [13], [15] across multiple end-users.

c) Algorithms and Baselines: We compare SOFA-I and SOFA-II against a series of baseline approaches. GLOBAL CRITERION and NEUTRAL COMPROMISE are classical no-preference methods [**?**], that seek to find a single solution. A key difference to our methods is that they do not consider an error in cost such as regret, but rather a normalized metric distance to a utopian, *i.e.,* idealistic, solution. EXPECTED is a uninformed 'out-of-the-box' setting where $\boldsymbol{w}^0 = [1/n, \dots, 1/n]$ while MEAN takes the average when a set of weights is given. This approach is commonly used when estimating weights that represent user preferences [13], [18]. Lastly, SOFA-SAMPLED first samples the entire Pareto front using the state-of-the-art MRPS algorithm from [4] and then selects the best solutions to Problem 1.

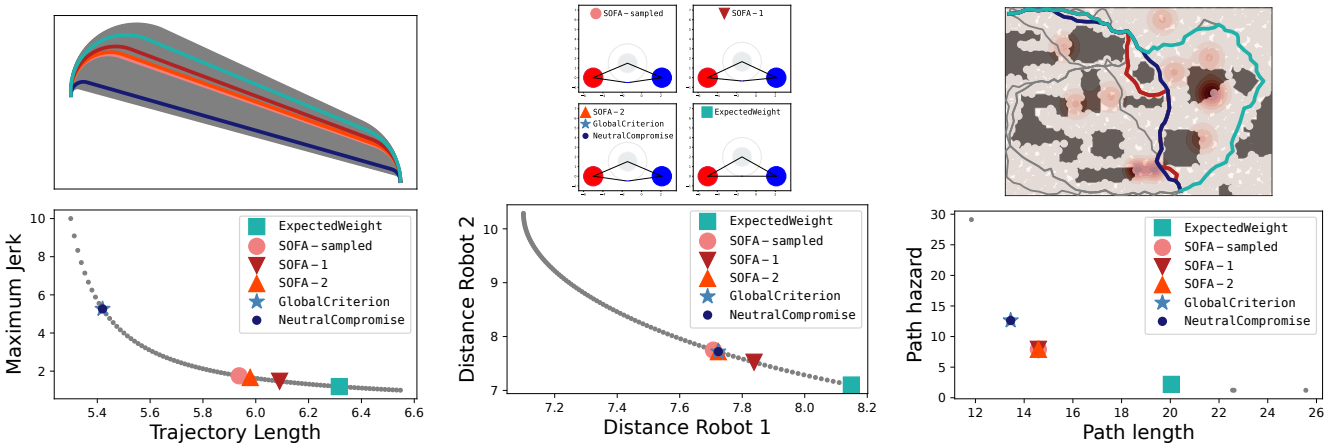
GLOBAL CRITERION, NEUTRAL COMPROMISE, EXPECTED, and MEAN are using only one call to the solver of the planning problem. The proposed SOFA-I and SOFA-II as well as SOFA-SAMPLED take a computation budget K as an input, allowing for K calls to the solver to (locally) explore the Pareto front. We show results for $K = 1, \dots, 20$.

B. Experiment 1— Algorithm effectiveness and efficiency

a) Qualitative Results: We begin with a qualitative analysis, highlighting an example for the solutions found by the different algorithms and the respective Pareto fronts for the three bi-objective planning problem, shown in Figures 3. These examples used computation budgets $K = 10$.

The DUBINS2D problem exhibits a continuous Pareto front. We observe that the different methods fall into two parts: GLOBAL CRITERION and NEUTRAL COMPROMISE find solutions of relatively short length with tight turning radius. In contrast, the other methods find similar solutions with roughly half the maximum jerk while the length only increases moderately. Interestingly, SOFA-I, SOFA-II and SOFA-SAMPLED do not find the same solution; indeed, SOFA-II finds the solution with lowest maximum regret.

Considering DISK SWAP, we notice a stark difference in the relative behaviour of compared methods. Here, SOFA-II, GLOBAL CRITERION and NEUTRAL COMPROMISE find the same solution with best maximum regret, while SOFA-SAMPLED performs comparably. In contrast, EXPECTED finds an extreme solution at the end of the Pareto front and SOFA-I finds an intermediate solution. It begins at the center weight $\boldsymbol{w}^0 = [1/n, \dots, 1/n]$, equivalent to EXPECTED, and improves via gradient search. Yet, the final solution suggests



(a) DUBINS2D. **(b)** DISK SWAP. **(c)** HAZARD NAVIGATION.

Fig. 3: Ground set of trajectories, example one-size-fits-all solutions and Pareto fronts for the three bi-objective planning problems

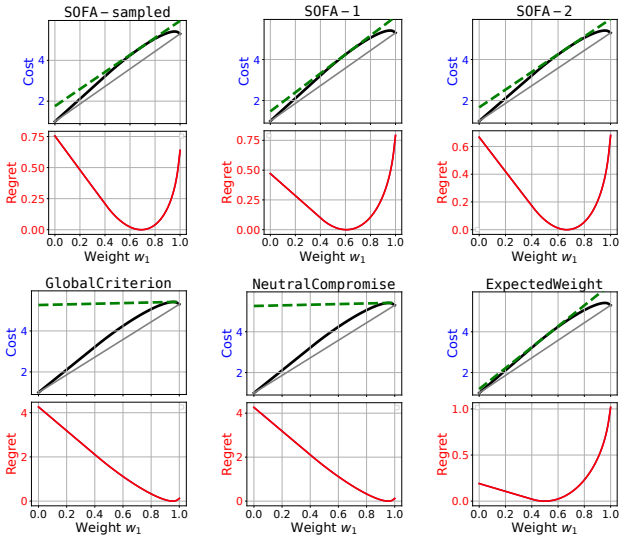


Fig. 4: Approximations of optimal cost and regret for DUBINS2D.

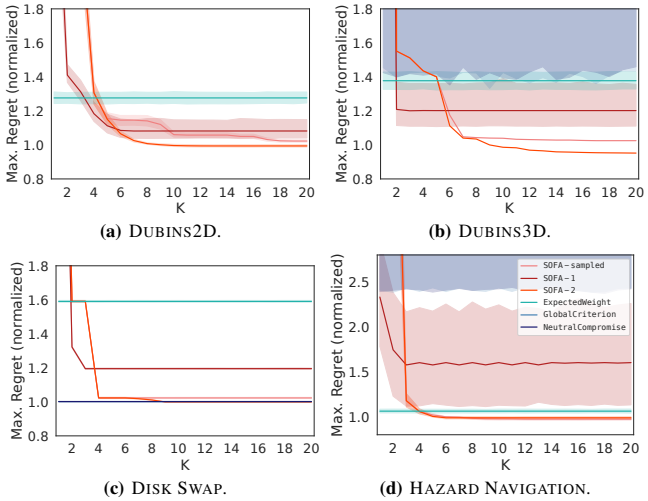
that it did not converge within the $K = 10$ iterations and thus is not as efficient as SOFA-II.

Finally, HAZARD NAVIGATION exhibits a discrete Pareto front with only six different solutions. Here, the different methods attain distinct solutions with NEUTRAL COMPROMISE and GLOBAL CRITERION resulting in the shortest paths with highest risk, the variants of our proposed method attain the same intermediate solution and EXPECTED results in a longer solution. Interestingly, SOFA-I converges to weight $[0.54, 0.46]$ and is thus close to the weight $[0.5, 0.5]$ for EXPECTED, yet results in a drastically different solution. Meanwhile, SOFA-II and SOFA-SAMPLED both converge to weight $[0.67, 0.33]$, and nonetheless attain the same solution as SOFA-I. This highlights that the problem of weight indefiniteness is particularly prevalent in discrete planning problems, as previously observed in [21].

We provide further insights into the characteristics of final solutions in Figure 4, for the example DUBINS2D instance shown in Figure 3a. The two-dimensional weight space $\mathcal{W}$ can be plotted as a single dimension since $w_1 + w_2 = 1$. In each subfigure, the upper plot shows the optimal cost function $u(\boldsymbol{w})$ (black), the interpolation plane L underneath (grey), and the tangent plane corresponding to the cost vector $\boldsymbol{f}$ of the solution returned by each method (green). The lower plot shows the regret over all weights in $\mathcal{W}$ (red).

For SOFA-SAMPLED, SOFA-I and SOFA-II, we observe almost identical solutions where the tangents are parallel to L. Further, the regret is minimal at the solution weight and increases towards both ends of $\mathcal{W}$, attaining a maximum of $\approx .75$. While SOFA-SAMPLED and SOFA-I show a slightly higher regret for $w_1 = 1$, the solutions for SOFA-II exhibit the same regret at both end-points, and is thus optimal. In contrast, GLOBAL CRITERION and NEUTRAL COMPROMISE find tangent planes that are *level*. While this implies that the incurred cost $h(s, \boldsymbol{w})$ remains constant over $\mathcal{W}$, the regret for $w_1 = 0$ is an order of magnitude higher than for $w_1 = 1$. Lastly, the tangent for EXPECTED is more similar to SOFA-II, but still attains regret that differ at the two endpoints, with maximum regret of ≈ 1 for $w_1 = 1$.

b) Quantitative Results: Next, we examine the maximum regret of solutions over the computation budget K, summarized in Figure 5. To compare regret between randomized instances, we normalize to an optimal solution to Problem 1 obtained by running SOFA-SAMPLED for $K = 100$ iterations. Since EXPECTED, GLOBAL CRITERION and NEUTRAL COMPROMISE are single shot methods, their regret is constant over K (absent lines are outside of the plots' competitive range). All three SOFA variants have a high regret for the first n iterations where the basis solutions are computed. However, within 3–4 iterations they outperform the strongest respective baseline. An exception is DISK SWAP where GLOBAL CRITERION and NEUTRAL COMPROMISE perform optimally; here SOFA-II attains the same solution after 8 iterations. SOFA-I makes quick progress initially, but levels out after $\approx$6 iterations. SOFA-SAMPLED finds better solutions overall, but also does not improve further after 10 iterations, DUBINS2D excepted. Since this approach reconstructs the entire Pareto front, newly added samples improve the approximation of the Pareto front, but do not necessarily the one-size-fits-all solution. In contrast, SOFA-II finds close to optimal solutions within 8 iterations



(a) DUBINS2D. **(b)** DUBINS3D. **(c)** DISK SWAP. **(d)** HAZARD NAVIGATION.

Fig. 5: Quantitative results for Experiment 1—Alg. efficiency.

for the bi-objective problems. In DUBINS3D the progress is slower due to higher complexity of an additional dimension. Nonetheless, SOFA-II continually improves and converges to minimum regret after 18 iterations.

In summary, the baselines are usually poor at addressing Problem 1, though GLOBAL CRITERION in DISK SWAP in an exception. While all three SOFA variants outperform the baselines, SOFA-II is most efficient, requiring fewest calls to a solver for the scalarized problem, and is the only method that converges to an optimal solution in all setups.

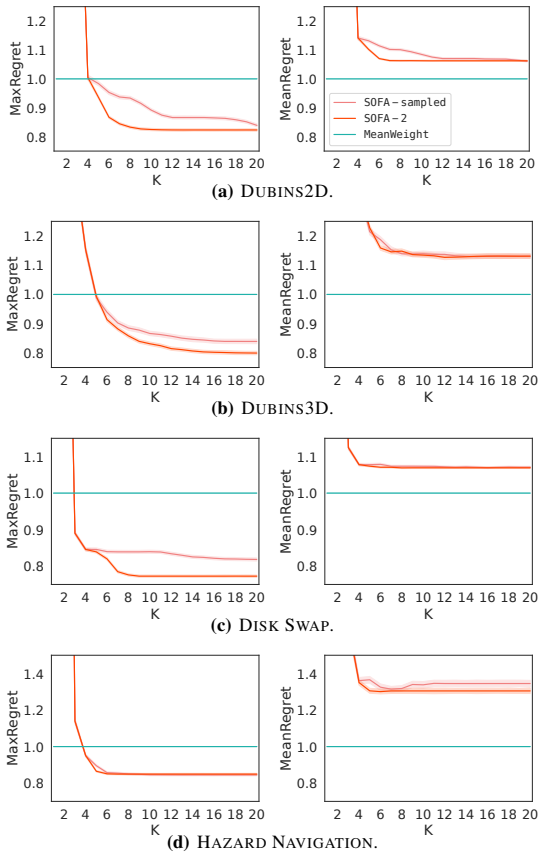
C. Experiment 2— One-size-fits-all for multiple users

In Experiment 2 we seek to find a one-size-fits-all solution among a randomly chosen set of user preferences, *i.e.,* a set of weights U. Building on our insights from Experiment 1, we only compare SOFA-II, SOFA-SAMPLED against the MEAN baseline. In the context of reward learning, expectations or averages are commonly used [15]. We employ two measures: the maximum regret and mean over U; both measures are normalized to the value attained by MEAN.

Results are shown in Figure 6. Similar to other baselines, the MEAN is a single shot method and thus remains constant. Throughout all four planning problems, we observe that SOFA approaches achieve a substantially lower maximum regret after only 3 iterations, and attain 75–85% of the maximum regret of MEAN after 10 iterations. Again, SOFA-II is more sample efficient than SOFA-SAMPLED in three of the four problems. On the other hand, the mean regret of SOFA is $\pm$10–40% higher compared to MEAN. A robust single weight is found to ensure that no user would be especially hard done by some selection that optimizes the preference of some hypothetical 'average' user. Thus, the proposed SOFA approaches are suitable when finding compromises between different user preferences that prioritize the worst-case performance, at the cost of the average—a decision which is likely fairer in preference-diverse user populations.

D. Key takeaways

In summary, the experiments show that SOFA-I and SOFA-II effectively tackle Problem 1. Notice that SOFA-II consistently performs comparable or superior to SOFA-SAMPLED with fewer samples. Given that SOFA-II builds upon the Pareto-exploration method in [4], which underpins SOFA-SAMPLED itself, it is remarkable to observe that the modification results in such greatly enhanced sample efficiency.



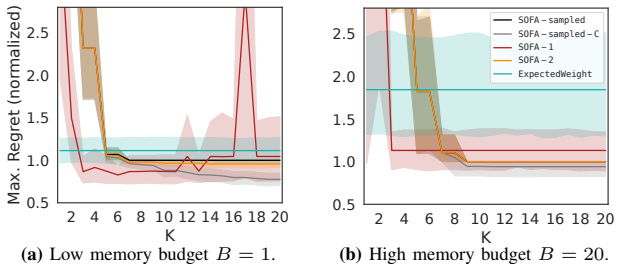
(a) DUBINS2D. **(b)** DUBINS3D. **(c)** DISK SWAP. **(d)** HAZARD NAVIGATION.

Fig. 6: Experiment 2—One-size-fits-all for multiple users.

V. DISCUSSION OF LIMITATIONS

The development in Section II explicitly introduced two simplifying assumptions: the first being non-multiplicity of solutions per weight, and the second being the restriction to linear scalarization. The second has implications that run far deeper: Pareto fronts of all the planning instances examined in this paper are convex and it is known that linear scalarization does not overlook solutions for such cases. However, when the Pareto front is non-convex, linear scalarization may miss non-dominated candidates [18]. Yet, linear scalarization is widely popular in robot planning; and it is important to note that convexity of the Pareto front does not imply convexity of the linearly scalarized objective function. Alternative methods (e.g., Chebyshev scalarization) can resolve this, but minimizing the objective (7) has not been justified under different scalarization functions.

At present, while the experiments reliably demonstrate a leveling-off of regret as K increases (see Figure 5), we have made no claims about convergence and no theoretical guarantees on convergence have been established. Also, both algorithms do assume access to an exact solver for the scalarized problem in (2): if the solver provides a suboptimal result, this would impact subsequent steps, altering the final weight



(a) Low memory budget $B = 1$. **(b)** High memory budget $B = 20$.

Fig. 7: Suboptimal solvers for complex path search, comparing a heuristic with memory budgets $B = 1$ and $B = 20$.

produced. We briefly analyze the case when the assumption is violated. We modify the HAZARD NAVIGATION setup to consider the maximum hazard opposed to cumulative one, such that the second objective is maximized over edges in the graph. This can no longer be solved optimally with a Dijkstra-style algorithm. Similar to the work of [18], we use a heuristic approach that stores multiple paths to every vertex, up to some memory budget B. Figure 7 shows results. In addition to earlier baselines, we include the SOFA-SAMPLED-C, adapting the Pareto sampling approach MRPS-C from [4] expressly designed to consider a suboptimal solver. For $B = 1$, SOFA-II (orange) is initially strong, but stops making progress after 10 iterations. SOFA-SAMPLED-C is robust to suboptimal solvers and thus continues to improve the solution. For $B = 20$, both approaches settle after 10 iterations and have a much smaller gap. As one might expect, the optimality assumption is less limiting when a strong heuristic is available. In both settings SOFA-I performs unreliably with the suboptimal solutions yielding incorrect gradients and can severely misguide the search. In summary, when no optimal solver is available, the proposed method SOFA-II is still able to yield good solutions, but can suffer from local optima; a pitfall which is avoided by the SOFA-SAMPLED-C baseline. Directly adapting the techniques from SOFA-SAMPLED-C, i.e., MRPS-C from [4], has not been seen to yield any theoretical or substantial empirical improvement to SOFA-II. In the context of a MOOP, suboptimality induces an error in the scalar cost. However, this can affect the error in the objectives $f_1, \ldots, f_n$ differently and thus poses further challenges for a guided search on the Pareto front to solve OSFA. Exploring this is beyond the scope of our paper.

VI. CONCLUSION AND FUTURE WORK

This paper examines how to find effective parameters for weighting individual objectives in multi-objective optimization problems, as can be usefully applied to robot planning.

The key perspective is that it is crucial to retain the idea that obtaining a scalarization is an activity that resolves uncertainty by considering the range of other, local potentialities. We introduced a notion of parameter robustness, expressed in terms of minimum regret, and proposed practical algorithms that can identify a robust weighting for a subclass of settings. Future work might increase this sub-class, and provide stronger theoretical underpinnings (as expressed in Section V). One might handle a broader class of problems via an approach that splits regions in the weight space, for instance, selecting splitting points based on maximally reducing regret.

A key motivation for this work was to try to understand the shortcomings arising from preference indefiniteness, and how they might be mitigated. We demonstrated (in Experiment 2) how regret may yield weightings that are 'fairer' in the sense of not punishing individuals with outré preferences excessively. An interesting question is how well weights transfer not merely across individuals, but to new scenarios.

REFERENCES

[1] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, "Defining and characterizing reward gaming," *NeurIPS*, pp. 9460–9471, 2022.

[2] I. R. T. C. for Model-Based Optimization for Robotics, "Optimization Tools," 2025, last access: 20-Jul-2025. [Online]. Available: https://www.ieee-ras.org/model-based-optimization-for-robotics/resources/optimization-tools

[3] "RSS'25 Workshop on Multi-objective optimization and planning in Robotics," Jun. 2025, last access: 1-Sep-2025. [Online]. Available: https://sites.google.com/view/moo-rss25/home

[4] A. Botros, N. Wilde, A. Sadeghi, J. Alonso-Mora, and S. L. Smith, "Regret-based sampling of pareto fronts for multi-objective robot planning problems," *IEEE Tr. Robotics*, vol. 40, pp. 3778–3794, 2024.

[5] S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, 2006.

[6] O. Salzman, A. Felner, C. Hernández, H. Zhang, S.-H. Chan, and S. Koenig, "Heuristic-Search Approaches for the Multi-Objective Shortest-Path Problem: Progress and Research Opportunities," in *IJCAI*, 2023, pp. 6759–6768.

[7] R. Wein, J. P. van den Berg, and D. Halperin, "The visibility-Voronoi complex and its applications," *Computational Geometry*, vol. 36, no. 1, pp. 66–87, 2007.

[8] B. Sakçak and S. M. LaValle, "Complete Path Planning That Simultaneously Optimizes Length and Clearance," in *Proc. of ICRA*, 2021.

[9] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation," *Structural and multidisciplinary optimization*, vol. 31, no. 2, pp. 105–116, 2006.

[10] S. Parisi, M. Pirotta, and J. Peters, "Manifold-based multi-objective policy search with sample reuse," *Neurocomputing*, vol. 263, pp. 3–14, 2017.

[11] B. W. Knox, S. Hatgis-Kessell, S. Booth, S. Niekum, P. Stone, and A. Allievi, "Models of human preference for learning reward functions," *Trans. Machine Learning Research*, 2024.

[12] S. Habibian, A. Jonnavittula, and D. P. Losey, "Here's what I've learned: Asking questions that reveal reward learning," *ACM Trans. on HRI*, vol. 11, pp. 1–28, 2022.

[13] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh, "Asking easy questions: A user-friendly approach to active reward learning," in *Proc. of CoRL*, 2020, pp. 1177–1190.

[14] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *Proc. of IROS*, 2011, pp. 1986–1993.

[15] N. Wilde, E. Biyik, D. Sadigh, and S. L. Smith, "Learning reward functions from scale feedback," in *Proc. of CoRL*, 2022, pp. 353–362.

[16] R. T. Clemen, *Making hard decisions: an introduction to decision analysis*. Duxbury Press Belmont, CA, 1996, vol. 2.

[17] T. Chugh, "Scalarizing functions in Bayesian multiobjective optimization," in *IEEE Congress on Evolutionary Computation*, 2020, pp. 1–8.

[18] N. Wilde, S. L. Smith, and J. Alonso-Mora, "Scalarizing multi-objective robot planning problems using weighted maximization," *IEEE RA-L*, vol. 9, no. 3, pp. 2503–2510, 2024.

[19] A. Botros, A. Sadeghi, N. Wilde, J. Alonso-Mora, and S. L. Smith, "Error-Bounded Approximation of Pareto Fronts in Robot Planning Problems," in *Proc. of WAFR*. Springer, 2022, pp. 506–522.

[20] H. Bauer, "Minimalstellen von Funktionen und Extremalpunkte," *Archiv def Mathematik*, vol. 9, pp. 389–393, 1958.

[21] N. Wilde, D. Kulić, and S. L. Smith, "Bayesian active learning for collaborative task specification using equivalence regions," *IEEE RA-L*, vol. 4, no. 2, pp. 1691–1698, 2019.

[22] D. Kirkpatrick and P. Liu, "Characterizing minimum-length coordinated motions for two discs," in *Proc. of CCCG*, 2016, pp. 252–259.