Scalarisation-based risk concepts for robust multi-objective optimisation

Ben Tu[*]   Nikolas Kantas[*]   Robert M. Lee[†]   Behrang Shafei[†]

Abstract

Robust optimisation is a well-established framework for optimising functions in the presence of uncertainty. The inherent goal of this problem is to identify a collection of inputs whose outputs are both desirable for the decision maker, whilst also being robust to the underlying uncertainties in the problem. In this work, we study the multi-objective case of this problem. We identify that the majority of all robust multi-objective algorithms rely on two key operations: robustification and scalarisation. Robustification refers to the strategy that is used to account for the uncertainty in the problem. Scalarisation refers to the procedure that is used to encode the relative importance of each objective to a scalar-valued reward. As these operations are not necessarily commutative, the order that they are performed in has an impact on the resulting solutions that are identified and the final decisions that are made. The purpose of this work is to give a thorough exposition on the effects of these different orderings and in particular highlight when one should opt for one ordering over the other. As part of our analysis, we showcase how many existing risk concepts can be integrated into the specification and solution of a robust multi-objective optimisation problem. Besides this, we also demonstrate how one can principally define the notion of a robust Pareto front and a robust performance metric based on our "robustify and scalarise" methodology. To illustrate the efficacy of these new ideas, we present two insightful case studies which are based on real-world data sets.

1   Introduction

Robust multi-objective optimisation is concerned with the problem of optimising a vector-valued function under uncertainty. In this paper, we will consider optimising a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, which is defined over a D-dimensional input space $\mathbb{X} \subseteq \mathbb{R}^D$ and a W-dimensional uncertain parameter space $\Xi \subseteq \mathbb{R}^W$. Here, the inputs $\mathbf{x} \in \mathbb{X}$ are used to denote all the parameters that can be controlled by the decision maker. The uncertain parameters $\boldsymbol{\xi} \in \Xi$ are used to denote all the other variables in the problem, which affect the value of the objective function, but are subject to uncertainty or randomness. Notably, these uncertain parameters might not be controllable. As a result, the goal of robust multi-objective optimisation is to identify a collection of controllable inputs whose distribution of output vectors are in some sense robust to the potential variation in these uncertain parameters. That is, we aim to solve the problem

$$\underset{\mathbf{x} \in \mathbb{X}}{\text{robustmax}}\, f(\mathbf{x}, \cdot) \tag{1}$$

where the 'robustmax' operator is some sensible robust generalisation of the vector-valued maximum which has to be defined by the decision maker.

---

[*]Imperial College London, United Kingdom
[†]BASF SE, Germany

The majority of existing work in robust multi-objective optimisation has largely focussed on defining the robust optimum via a first principles Pareto ordering style approach where one defines a partial ordering over the space of inputs—see for instance the work by Ide and Köbis [2014], Ide and Schöbel [2016] and Botte and Schöbel [2019] for a review. Conceptually, this partial ordering determines whether an input $\mathbf{x} \in \mathbb{X}$ is considered to be more robust than another input $\mathbf{x}' \in \mathbb{X}$ with respect to the objective function f. The solution to the corresponding robust optimisation problem is then given by the collection of inputs which are not robustly dominated by any other feasible input. Consequently, one is then typically interested in coming up with a principled computational strategy in order to identify these robust points. The most prevalent approach to accomplish this is the scalarisation strategy. In this approach, one recasts the corresponding robust multi-objective optimisation problem into a sensible collection of scalar-valued optimisation problems; these problems can then be solved using traditional techniques in order to obtain an approximation to the robust set of interest. Although theoretically motivated, the first principles approach is rather rigid in practice because whenever we want to use a new partial ordering, we would have to manually identify and derive a new set of results. In this work, we take a more efficient approach to robust multi-objective optimisation and define robustness directly using scalarisation functions. That is, we define the solution of the robust multi-objective optimisation (1) as being equivalent to solving a collection of general robust scalarised problems that can be readily solved using standard scalar-valued optimisation techniques.

There are the two essential ingredients to our robust multi-objective optimisation strategy: scalarisation and robustification. Scalarisation refers to the procedure of mapping a vector-valued objective function to a family of scalar-valued rewards. Meanwhile, robustification is concerned with the process of quantifying the robust value of a set of possible points. In this work, we focus on the use of risk functionals, otherwise known as risk measures, in order to perform the robustification procedure. These risk-inspired concepts of robustness are very well established in the robust single-objective optimisation literature [Quaranta and Zaffaroni, 2008, Ben-Tal et al., 2009, Bertsimas et al., 2011, Rahimian and Mehrotra, 2019], but have not been adequately studied in the multi-objective setting. Our work aims to address this gap by providing a generic methodology for general risk functionals. Similarly, for the scalarisation procedure, our methodology is general and can accommodate for any arbitrary family of scalarisation functions such as the Gerstewitz functional [Gerstewitz, 1983, Tammer and Weidner, 2020]. Nevertheless, as a working example, we will often use the length scalarisation functions. These are a variant of the Chebyshev scalarisation functions that are known to exhibit many intuitive and useful properties which will be discussed throughout [Ishibuchi et al., 2009, Deb and Jain, 2012, Tu et al., 2024].

As identified in preceding papers [Fliege and Werner, 2014, Groetzner and Werner, 2022], the robustification and scalarisation operations are not commutative in general: the order matters. A core part of our work is a careful analysis of when one should robustify then scalarise (RTS) or whether one should scalarise then robustify (STR). We will highlight later in Section 3.3.2, that one should take an RTS approach in the many-time aggregating scenario where one is interested in the aggregated performance at an input over many repeated runs. In contrast, one should take an STR approach in the one-time or non-aggregating scenario where one wishes to obtain consistently strong performance on every individual run.

1.1 Main contributions

As part of this work, we have made several novel and insightful contributions to the study of robust multi-objective optimisation. Firstly, we have introduced and formalised two general

computationally efficient approaches to robust multi-objective optimisation: the RTS approach (Section 3.1) and the STR approach (Section 3.2). As we motivate in Section 3.3.1, these two approaches are very flexible and they encompass many existing robust multi-objective solution strategies as special cases. Secondly, we present a detailed discussion on when one should use an RTS or STR approach (Section 3.3.2) and when these approaches are equivalent (Section 3.3.3). Thirdly, inspired by the recent results by Tu et al. [2024], we showcase in Section 4 how it is possible to define a valid Pareto front surface which reflects the robust trade-off that is happening under both the RTS and STR paradigm for the length scalarised problems. Specifically, we derive the novel concept of the RTS front (Definition 4.3) and STR front (Definition 4.4), which are polar surfaces described by the robust length scalarised values. Fourthly, in Section 5, we present a general class of performance metrics that can be used under the RTS and STR setting in order to assess the performance of any robust multi-objective algorithm. In particular, we generalise the construction of the R2 utility [Hansen and Jaszkiewicz, 1998] to the RTS and STR setting.

1.2   Structure of the paper

The rest of the paper is organised as follows: In Section 2, we recall the basics of multi-objective optimisation (Section 2.1), the scalarisation strategy (Section 2.2) and risk functionals (Section 2.3). We then move on to Section 3, where we first introduce the RTS approach (Section 3.1) and STR approach (Section 3.2) and then discuss their relation with existing work (Section 3.3). Afterwards, we move on to Section 4, where we introduce two generalisations of the Pareto front surface, namely the RTS front and STR front. Besides this, in Section 5, we introduce the robust R2 utilities, which can be used to quantify the performance of any approximate robust Pareto set. Consequently, in Section 6, we present some numerical experiments which showcases how our methodology can be used in practice on some real-world data sets. Finally, in Section 7, we conclude the paper with a short summary and discussion. All of the proofs for the main results in this paper are presented in Appendix A, whilst the scripts that are needed to reproduce the figures and experiments are available in our Github repository: https://github.com/benmltu/scalarize.

2   Preliminaries

2.1   Multi-objective optimisation

Given a bounded vector-valued objective function $g : \mathbb{X} \to \mathbb{R}^M$, defined over some D-dimensional input space $\mathbb{X} \subseteq \mathbb{R}^D$, the standard multi-objective optimisation problem is

$$\max_{\mathbf{x} \in \mathbb{X}} g(\mathbf{x}), \tag{2}$$

where the maximum here is defined using the Pareto partial ordering relations, which we recall below. Note that throughout, we will use $g(\mathbf{x}) \in \mathbb{R}^M$ to denote a generic vector-valued objective; we will use $f(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{R}^M$ when the objective is also dependent on an uncertainty parameter $\boldsymbol{\xi} \in \Xi$. In addition, we will focus entirely on maximisation problems. This can be done without loss of generality as $\min(A) = -\max(-A)$ for any set of vectors $A \subset \mathbb{R}^M$.

Definition 2.1 (Pareto domination) *The weak, strict and strong Pareto domination are denoted by the binary relations $\succeq, \succ$ and $\succ\!\!\!\succ$, respectively. We say a vector $\mathbf{y} \in \mathbb{R}^M$ weakly,*

3

strictly or *strongly Pareto dominates another vector* $\mathbf{y}' \in \mathbb{R}^M$ *if*

$$\mathbf{y} \succeq \mathbf{y}' \iff \mathbf{y} - \mathbf{y}' \in \mathbb{R}_{\geq 0}^M,$$
$$\mathbf{y} \succ \mathbf{y}' \iff \mathbf{y} - \mathbf{y}' \in \mathbb{R}_{\geq 0}^M \setminus \{\mathbf{0}_M\},$$
$$\mathbf{y} \succ\succ \mathbf{y}' \iff \mathbf{y} - \mathbf{y}' \in \mathbb{R}_{>0}^M,$$

respectively, where $\mathbf{0}_M \in \mathbb{R}^M$ *denotes the* M*-dimensional vector of zeros.*

Definition 2.2 (Domination region) *The Pareto domination (or dominated) region of a set of vectors* $A \subseteq \mathbb{R}^M$ *is defined as the collection of vectors which dominates (or is dominated by) this set, that is*

$$\mathbb{D}_\diamond[A] := \cup_{\mathbf{a} \in A} \{\mathbf{y} \in \mathbb{R}^M : \mathbf{y} \diamond \mathbf{a}\},$$

where $\diamond \in \{\succeq, \succ, \succ\succ\}$ *(or* $\diamond \in \{\preceq, \prec, \prec\prec\}$*) denotes a partial ordering relation. In addition, we denote the complement of any domination region by* $\mathbb{D}_\diamond^C[A] := \mathbb{R}^M \setminus \mathbb{D}_\diamond[A]$.

Definition 2.3 (Pareto optimality) *Given a bounded set of vectors* $A \subset \mathbb{R}^M$, *a point* $\mathbf{a} \in A$ *is weakly or strictly Pareto optimal if we cannot find another vector* $\mathbf{a}' \in A$ *which strongly or strictly Pareto dominates it, respectively. The collection of all of the weakly or strictly Pareto optimal points in this set is referred to as the weak or strict Pareto front,* $\mathcal{Y}^{\text{weak}}[A] := A \cap \mathbb{D}_{\prec\prec}^C[A]$ *and* $\mathcal{Y}^{\text{strict}}[A] := A \cap \mathbb{D}_{\prec}^C[A]$, *respectively.*

Solving the standard multi-objective optimisation problem (2) is equivalent to finding the corresponding optimal set of inputs which maps to the Pareto front of interest. This set of optimal inputs is commonly referred to as the Pareto set. In other words, the standard goal of interest in vector-valued optimisation is to target either the weak Pareto set or the strict Pareto set: $\mathcal{X}_g^{\text{weak}} := g^{-1}(\mathcal{Y}^{\text{weak}}[g(\mathbb{X})])$ or $\mathcal{X}_g^{\text{strict}} := g^{-1}(\mathcal{Y}^{\text{strict}}[g(\mathbb{X})])$, respectively, where $g(\mathbb{X}) := \{g(\mathbf{x}) \in \mathbb{R}^M : \mathbf{x} \in \mathbb{X}\}$ is the image of the objective function.

2.2 Scalarisation perspective

The standard strategy that is often used to solve (2) is a scalarisation method. In this approach, we recast the vector-valued optimisation problem of interest into a collection of scalar-valued optimisation problems

$$\{\max_{\mathbf{x} \in \mathbb{X}} s_{\boldsymbol{\theta}}(g(\mathbf{x})) : \boldsymbol{\theta} \in \Theta\}, \tag{3}$$

where $s_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}$ is a parametric scalarisation function that is indexed by scalarisation parameters $\boldsymbol{\theta} \in \Theta$. Notably, if one is interested in capturing the Pareto front using (3), then this would require us to define a suitable[1] set of scalarisation functions $\{s_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ which ensures that the resulting solutions are Pareto optimal. For instance, this former property is usually assured by restricting our attention to scalarisation functions that are monotonically increasing [Tu et al., 2025, Definition 2.5]. In practice however, a decision maker might not necessarily be interested in just targetting the Pareto front. In particular they might be open to the use of non-monotonic scalarisation functions, which can be used to target desirable points that are not necessarily Pareto optimal. For example, they might consider the use of distance-based scalarisation functions in order to target a set of desired specifications.

Example 2.1 (Lp scalarisation function) *If a decision maker could elicit a desired target vector* $\boldsymbol{v} \in \mathbb{R}^M$, *then they might consider setting the scalarisation function to be the negative*[2]

---

[1]In theory, one might desire a family of scalarisation functions that can target all Pareto optimal solutions. In practice however, one can usually only recover a finite approximation to the optimal set and therefore this requirement of complete coverage might not be practically necessary.

[2]The negative sign is included because we want the points closer to the target vectors to have a higher scalarised value.
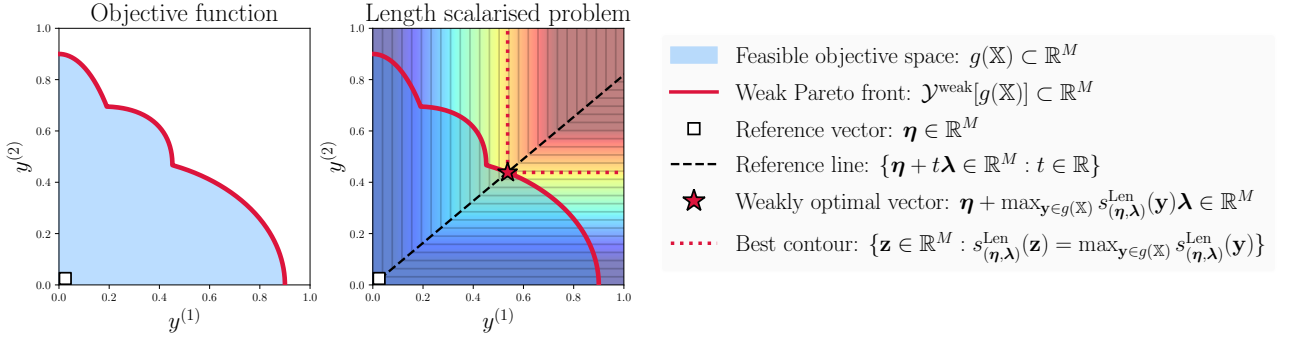
Figure 1: *An illustration of a length scalarised problem in $M = 2$ dimensions.*

weighted L^p distance to this point:

$$s^{\mathrm{Lp}}_{(\boldsymbol{v},\mathbf{w})}(\mathbf{y}) = -\Big(\sum_{m=1}^{M}(w^{(m)}(v^{(m)} - y^{(m)}))^p\Big)^{1/p} \tag{4}$$

for some weight vector $\mathbf{w} \in \Delta^{M-1} := \{\mathbf{y} \in \mathbb{R}^M_{\geq 0} : ||\mathbf{y}||_{L^1} = 1\}$. Note that the solution to the corresponding scalarised optimisation problem (3) is not necessarily Pareto optimal if the target vector lies in the feasible space. Specifically, the scalarisation function (4) would not be monotonic over the feasible space and it would actually penalise objective vectors $g(\mathbf{x}) \in \mathbb{R}^M$ which Pareto dominate the target vector $\boldsymbol{v}$.

As a working example, we will often use the length scalarisation function, which is a specific reparameterisation of the Chebyshev scalarisation function [Ishibuchi et al., 2009, Deb and Jain, 2012, Tu et al., 2024]:

$$s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(\mathbf{y}) = \min_{m=1,\dots,M} \frac{\max(y^{(m)} - \eta^{(m)}, 0)}{\lambda^{(m)}} \tag{5}$$

for any vector $\mathbf{y} \in \mathbb{R}^M$, where $\boldsymbol{\eta} \in \mathbb{R}^M$ denotes a fixed reference vector and $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$ denotes a positive unit vector that lives in the space of all positive unit vectors $\mathcal{S}^{M-1}_+ := \{\mathbf{y} \in \mathbb{R}^M_{>0} : ||\mathbf{y}||_{L^2} = 1\}$. Conceptually, for any vector $\mathbf{y}$ that strongly dominates the reference vector $\boldsymbol{\eta}$, the length scalarisation function returns the projected length of this vector along the line $\{\boldsymbol{\eta} + t\boldsymbol{\lambda} : t \in \mathbb{R}\}$, that is, $s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(\mathbf{y}) = \max\{t > 0 : \boldsymbol{\eta} + t\boldsymbol{\lambda} \preceq \mathbf{y}\}$ for any $\mathbf{y} \in \mathbb{D}_{\gg}[\{\boldsymbol{\eta}\}]$—see the work by Tu et al. [2024] for more details for this construction. We now recall a key result from the multi-objective optimisation literature, which states that all weakly Pareto optimal solutions of (2) can be recovered by solving a set of length scalarised problems (3). For some additional intuition, we have also included Figure 1, which showcases the geometric idea behind this result in two dimensions.

Theorem 2.1 (Weak solutions) *[Miettinen, 1998, Part 2, Theorem 3.4.5] Consider a bounded objective function $g : \mathbb{X} \to \mathbb{R}^M$, and a reference vector $\boldsymbol{\eta} \in \cap_{\mathbf{x} \in \mathbb{X}} \mathbb{D}_{\prec\!\prec}[\{g(\mathbf{x})\}]$ which is strongly dominated by the whole feasible objective space. Then, for any weakly Pareto optimal input $\mathbf{z} \in \mathcal{X}^{\mathrm{weak}}_g$, there exists a positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$ such that the input $\mathbf{z}$ is a solution to the scalarised optimisation problem*

$$\max_{\mathbf{x} \in \mathbb{X}} s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(g(\mathbf{x})) = \max_{\mathbf{x} \in \mathbb{X}} \min_{m=1,\dots,M} \frac{g^{(m)}(\mathbf{x}) - \eta^{(m)}}{\lambda^{(m)}}. \tag{6}$$

Namely we can choose $\boldsymbol{\lambda} = (g(\mathbf{z}) - \boldsymbol{\eta})/||g(\mathbf{z}) - \boldsymbol{\eta}||_{L^2} \in \mathcal{S}^{M-1}_+$.
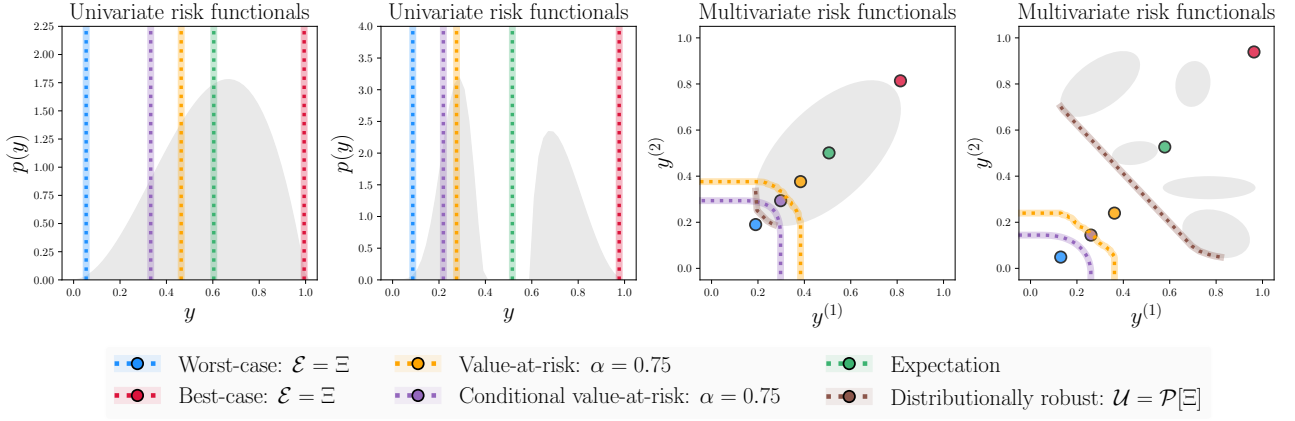
Figure 2: *An illustration of some univariate and multivariate risk functionals. On the left plots, we consider two univariate probability distributions and plot its associated risk-adjusted values. On the right plots, we consider two uniform distributed random variables and plot its associated risk-adjusted sets. In these multivariate plots, the circles are used to denote the component-wise risk functionals (Example 2.11), whilst the lines are used to denote the output of the set-valued risk functionals (Examples 2.9, 2.10 and 2.12).*

2.3 Risk functionals

In robust optimisation, one generally evaluates the value of an input according to the distribution of its objective values. That is, given an objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ the value at any input $\mathbf{x} \in \mathbb{X}$ is typically determined by its corresponding output set

$$f(\mathbf{x}, \Xi) := \{f(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{R}^M : \boldsymbol{\xi} \in \Xi\}. \tag{7}$$

Most existing work in robust multi-objective optimisation has focussed on the simple scenario where each uncertain parameter $\boldsymbol{\xi} \in \Xi$ is given equal weight. In this work, we consider a more general treatment where the uncertain parameter is treated as a random variable that is defined sensibly over a standard probability space. More precisely, we suppose that the uncertain parameter $\boldsymbol{\xi} \in \Xi$ is distributed according to some probability density $p \in \mathcal{P}[\Xi]$ that could potentially be known but might in general be unknown. Equipped with this set-up, the value of an input $\mathbf{x} \in \mathbb{X}$ will no longer be determined by its equally weighted output set (7), but instead will be defined using a sensible weighted alternative. More specifically, we will consider the use of risk functionals [Rüschendorf, 2013, Part 2] in order to obtain these proxy set of values. For this reason, we will now give a brief overview on the concepts of a univariate risk functional (Section 2.3.1) and a multivariate risk functional (Section 2.3.2). In Figure 2, we give an illustration of some univariate and multivariate risk functionals for a simple one-dimensional and two-dimensional example, respectively.

2.3.1 Univariate risk functionals

A univariate risk functional ρ is any function that maps any real-valued function $h : \Xi \to \mathbb{R}$ to a real-valued scalar $\rho[h] \in \mathbb{R}$. Intuitively, the value $\rho[h]$ summarises the quality of the output $h(\boldsymbol{\xi}) \in \mathbb{R}$ according to the uncertainty that is inherited from $\boldsymbol{\xi} \in \Xi$. In the following, we present examples of many popular univariate risk functionals that have appeared in the robust optimisation literature.

Example 2.2 (Extreme cases) *Given an uncertainty set $\mathcal{E} \subseteq \Xi$, the worst-case and best-case functional maps any function $h : \Xi \to \mathbb{R}$ to its corresponding infimum value $\rho^{\text{WorstCase}}[h] := \inf_{\boldsymbol{\xi} \in \mathcal{E}} h(\boldsymbol{\xi})$ and supremum value $\rho^{\text{BestCase}}[h] := \sup_{\boldsymbol{\xi} \in \mathcal{E}} h(\boldsymbol{\xi})$, respectively.*

Example 2.3 (Expectation) *Given a probability density $p \in \mathcal{P}[\Xi]$, the expectation functional maps any function $h : \Xi \to \mathbb{R}$ to its corresponding expectation $\rho^{\text{Exp}}[h] := \mathbb{E}_{p(\boldsymbol{\xi})}[h(\boldsymbol{\xi})]$.*

Example 2.4 (Value-at-risk) *Given a probability density $p \in \mathcal{P}[\Xi]$, the value-at-risk (VaR) functional, at level $\alpha \in (0, 1)$, maps any function $h : \Xi \to \mathbb{R}$ to its corresponding $(1-\alpha)$-quantile value*

$$\rho^{\mathrm{VaR}_\alpha}[h] := \sup\{y \in \mathbb{R} : \mathbb{E}_{p(\boldsymbol{\xi})}[\mathbb{1}[h(\boldsymbol{\xi}) \geq y]] \geq \alpha\}.$$

Example 2.5 (Distributionally robust) *Given an uncertainty set of probability densities $\mathcal{U} \subseteq \mathcal{P}[\Xi]$, the distributionally robust (DR) functional maps any function $h : \Xi \to \mathbb{R}$ to its distributionally worst-case expected value*

$$\rho^{\mathrm{DR}}[h] := \inf_{q \in \mathcal{U}} \mathbb{E}_{q(\boldsymbol{\xi})}[h(\boldsymbol{\xi})].$$

Note that if we set $\mathcal{U} = \mathcal{P}[\mathcal{E}]$, then we recover the worst-case functional $\inf_{q \in \mathcal{P}[\mathcal{E}]} \mathbb{E}_{q(\boldsymbol{\xi})}[h(\boldsymbol{\xi})] = \inf_{\boldsymbol{\xi} \in \mathcal{E}} h(\boldsymbol{\xi})$. As showcased in existing work [Shapiro, 2017, Rahimian and Mehrotra, 2019], we can also recover the conditional value-at-risk (CVaR) functional [Rockafellar and Uryasev, 2002] at level $\alpha \in (0, 1)$, with respect to a distribution $p \in \mathcal{P}[\Xi]$ as a special case of the DR functional

$$\rho^{\mathrm{CVaR}_\alpha}[h] := \frac{1}{1 - \alpha} \int_\alpha^1 \rho^{\mathrm{VaR}_\tau}[h] d\tau.$$

In the following sections, we will consider the problem of maximising the output of a risk functional. We will often refer to the resulting output of a univariate risk functional as the risk-adjusted or risk-penalised objective value. Naturally, we want these risk functionals to satisfy some sensible properties in order for the corresponding risk-penalised optimisation problem to make sense from a modelling perspective. For this reason, we focus our attention on partially coherent univariate risk functional (Definition 2.4). By design, these functionals satisfy four additional properties which are sufficient to prove many of the results which we state and describe later in Section 4. Note that all of the univariate risk functionals listed above (Examples 2.2 to 2.5) are partially coherent.

Definition 2.4 (Partial coherency) *A univariate risk functional ρ is partially coherent if it satisfies the following four properties:*

(i) *Normalised: if $h : \Xi \to \mathbb{R}$ is the zero function, that is $h(\boldsymbol{\xi}) = 0$ for all $\boldsymbol{\xi} \in \Xi$, then $\rho[h] = 0$.*

(ii) *Monotonicity: for any functions $a : \Xi \to \mathbb{R}$ and $b : \Xi \to \mathbb{R}$ such that $a(\boldsymbol{\xi}) \geq b(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \Xi$, then $\rho[a] \geq \rho[b]$.*

(iii) *Positively homogeneous: for any positive scalar $\alpha > 0$ and function $h : \Xi \to \mathbb{R}$, we have that $\rho[\alpha h] = \alpha \rho[h]$.*

(iv) *Translation equivariance: for any constant $c \in \mathbb{R}$ and function $h : \Xi \to \mathbb{R}$, we have that $\rho[h + c] = \rho[h] + c$.*

Remark 2.1 (Coherency) *A partially coherent univariate risk functional becomes a coherent univariate risk functional, as defined by Artzner et al. [1999], if it satisfies an additional property known as sub-additivity: $\rho[a + b] \leq \rho[a] + \rho[b]$ for any functions $a : \Xi \to \mathbb{R}$ and $b : \Xi \to \mathbb{R}$. By a well-known duality result [Artzner et al., 1999, Proposition 4.1], it is known that any coherent univariate risk functional can be written as a distributionally robust functional (Example 2.5) with some uncertainty set $\mathcal{U} \subseteq \mathcal{P}[\Xi]$.*

To prove some results later on, we also require that the univariate risk functional are bounded (Definition 2.5). It is not hard to show that any partially coherent univariate risk functional

is indeed bounded. For completeness, we state this result in Lemma 2.1 and prove it in Appendix A.2.

Definition 2.5 (Bounded risk) *A univariate risk functional ρ is bounded if for any bounded function $h : \Xi \to \mathbb{R}$, the risk-adjusted value $\rho[h] \in \mathbb{R}$ is bounded.*

Lemma 2.1 *A partially coherent univariate risk functional ρ is bounded.*

2.3.2 Multivariate risk functionals

A multivariate risk functional $\boldsymbol{\rho}$ is any function that maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to a closed set of vectors $\boldsymbol{\rho}[h] \subseteq \mathbb{R}^M$. Intuitively, the set of vectors $\boldsymbol{\rho}[h]$ acts as a risk-adjusted summary set of the output set $h(\Xi) \subseteq \mathbb{R}^M$. In the following, we present examples of many popular multivariate risk functionals that have appeared in the robust optimisation literature.

Example 2.6 (Identity) *The identity functional maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to its output set $\boldsymbol{\rho}^{\text{Identity}}[h] := \{h(\boldsymbol{\xi}) \in \mathbb{R}^M : \boldsymbol{\xi} \in \Xi\}$.*

Example 2.7 (Multivariate extreme cases) *Given an uncertainty set $\mathcal{E} \subseteq \Xi$, the multivariate worst-case and best-case functional [Kuroiwa and Lee, 2012, Fliege and Werner, 2014, Ehrgott et al., 2014] maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to the singleton set of its corresponding component-wise infimum and supremum value, respectively:*

$$\boldsymbol{\rho}^{\text{WorstCase}}[h] := \{(\inf_{\boldsymbol{\xi} \in \mathcal{E}} h^{(1)}(\boldsymbol{\xi}), \dots, \inf_{\boldsymbol{\xi} \in \mathcal{E}} h^{(M)}(\boldsymbol{\xi}))\},$$

$$\boldsymbol{\rho}^{\text{BestCase}}[h] := \{(\sup_{\boldsymbol{\xi} \in \mathcal{E}} h^{(1)}(\boldsymbol{\xi}), \dots, \sup_{\boldsymbol{\xi} \in \mathcal{E}} h^{(M)}(\boldsymbol{\xi}))\}.$$

Example 2.8 (Multivariate expectation) *Given a probability density $p \in \mathcal{P}[\Xi]$, the multivariate expectation functional maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to the singleton set comprised of its corresponding expectation $\boldsymbol{\rho}^{\text{Exp}}[h] := \{\mathbb{E}_{p(\boldsymbol{\xi})}[h(\boldsymbol{\xi})]\}$. For example, Deb and Gupta [2005] considered a special case where the uncertain parameter models any additive input noise arising in (2), that is $\boldsymbol{\rho}^{\text{Exp}}[f(\mathbf{x}, \cdot)] = \{\mathbb{E}_{p(\boldsymbol{\xi})}[g(\mathbf{x} + \boldsymbol{\xi})]\}$ for any input $\mathbf{x} \in \mathbb{X}$, where $p \in \mathcal{P}[\Xi]$ is the input noise distribution and $g : \mathbb{X} \to \mathbb{R}^M$ is the vector-valued objective function.*

Example 2.9 (Multivariate value-at-risk) *Given a probability density $p \in \mathcal{P}[\Xi]$, the multivariate value-at-risk [Prékopa, 2012], at a level $\alpha \in (0, 1)$, maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to the set*

$$\boldsymbol{\rho}^{\text{VaR}_\alpha}[h] := \sup\{\mathbf{y} \in \mathbb{R}^M : \mathbb{E}_{p(\boldsymbol{\xi})}[\mathbb{1}[h(\boldsymbol{\xi}) \succeq \mathbf{y}]] \geq \alpha\},$$

where the supremum of a set of vectors is defined with respect to the weak Pareto partial ordering on vectors. At the lowermost case, when $\alpha \downarrow 0$, we obtain the weak Pareto front of the domination region of the essential output set: $\mathcal{Y}^{\text{weak}}[\mathbb{D}_{\preceq}[h(\Xi^{\text{ess}})]]$ where $\Xi^{\text{ess}} := \text{ess supp}(p)$ is the essential support of p. In the uppermost case, when $\alpha \uparrow 1$, we obtain the weak Pareto front of the domination region of the worst-case vector: $\mathcal{Y}^{\text{weak}}[\mathbb{D}_{\preceq}[\boldsymbol{\rho}^{\text{WorstCase}}[h]]]$ where $\mathcal{E} = \Xi^{\text{ess}}$.

Example 2.10 (Multivariate distributionally robust) *Given an uncertainty set of probability densities $\mathcal{U} \subseteq \mathcal{P}[\Xi]$, the multivariate distributionally robust functional maps any vector-valued function $h : \Xi \to \mathbb{R}^M$ to its corresponding distributionally robust worst-case weak Pareto front*

$$\boldsymbol{\rho}^{\text{DR}}[h] := \inf_{q \in \mathcal{U}} \mathbb{E}_{q(\boldsymbol{\xi})}[h(\boldsymbol{\xi})],$$

where the inf *operator here is defined with respect to the weak Pareto partial ordering on vectors. A special case of this problem has arisen in the work by* Bokrantz and Fredriksson *[2017, Section 5], who studied the setting where* $\mathcal{U} = \mathcal{P}[\Xi]$. *They showed that the corresponding set of vectors is the weak Pareto minimal front of the convex hull of the output set:* $-\mathcal{Y}^{\text{weak}}[-\text{ConvexHull}[h(\Xi)]]$.

Example 2.11 (Component-wise risk) *Given a collection of bounded univariate risk functionals* $\{\rho^{(1)}, \ldots, \rho^{(M)}\}$, *the component-wise (CW) multivariate risk functional maps any vector-valued function* $h : \Xi \to \mathbb{R}^M$ *to the singleton set comprised of its corresponding component-wise risk-adjusted vector*

$$\boldsymbol{\rho}^{\text{CW}}[h] := \{(\rho^{(1)}[h^{(1)}], \ldots, \rho^{(M)}[h^{(M)}])\},$$

where $h(\boldsymbol{\xi}) = (h^{(1)}(\boldsymbol{\xi}), \ldots, h^{(M)}(\boldsymbol{\xi})) \in \mathbb{R}^M$ *for all* $\boldsymbol{\xi} \in \Xi$. *The multivariate extreme cases (Example 2.7) and multivariate expectation (Example 2.8) are special cases of this component-wise risk functional.*

Example 2.12 (Pareto front statistics) *Given a partially coherent univariate risk functional* ρ *and reference vector* $\boldsymbol{\eta} \in \mathbb{R}^M$, *the Pareto front surface risk statistic, which we define and derive later in Proposition 4.1, maps any vector-valued function* $h : \Xi \to \mathbb{R}^M$ *to its corresponding Pareto front risk statistic*

$$\boldsymbol{\rho}^{\rho\text{-statistic}}[h] := \rho[\mathcal{Y}^{\text{int}}_{\boldsymbol{\eta}}[\{h(\cdot)\}]] = \{\boldsymbol{\eta} + \rho[s^{\text{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(h(\cdot))]\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+\}.$$

We illustrate an instance of this multivariate risk functional in Figure 2 for the CVaR Pareto front surface statistic. Note that this multivariate notion of the CVaR is based on the univariate CVaR (Example 2.5) and is different from the multivariate generalisation described by Prékopa *[2012] which is much more challenging to define and compute.*

For some results later on, we require that the multivariate risk functional of interest is bounded (Definition 2.6). Notably, all of the multivariate risk functionals defined above (Examples 2.7 to 2.12) are bounded.

Definition 2.6 (Bounded risk) *A multivariate risk functional* $\boldsymbol{\rho}$ *is bounded if for any bounded vector-valued function* $h : \Xi \to \mathbb{R}^M$, *the risk-adjusted set* $\boldsymbol{\rho}[h] \subseteq \mathbb{R}^M$ *is a bounded set of vectors.*

3 Robust multi-objective optimisation

Several notions of robustness have already been proposed for the robust multi-objective optimisation problem (1). Regardless of these different definitions, the most prominent solution strategies that have been proposed to solve these robust problems are all largely based on the scalarisation methodology. We now proceed to introduce the two main solution strategies that will be the primary focus of this work: the robustify then scalarise (8) approach and the scalarise then robustify (11) approach.

3.1 Robustify then scalarise

We now introduce a general class of robust multi-objective optimisation problems which we refer to as the robustify then scalarise problems. A scalar-valued optimisation problem belongs to this class if it can be written in the form

$$\max_{\mathbf{x} \in \mathbb{X}} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]} s_{\boldsymbol{\theta}}(\mathbf{y}), \tag{8}$$

where $s_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}$ denotes any scalarisation function and $\boldsymbol{\rho}$ denotes a multivariate risk functional (Section 2.3.2). By design, an RTS problem associates each input $\mathbf{x} \in \mathbb{X}$ with its largest scalarised risk-adjusted value: $\max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]} s_{\boldsymbol{\theta}}(\mathbf{y}) \in \mathbb{R}$. The solution of an RTS problem is then given by the subset of inputs which map to the largest possible scalarised risk-adjusted value. We refer to these solutions as the RTS robust solutions of f (with respect to $s_{\boldsymbol{\theta}}$ and $\boldsymbol{\rho}$). Following the standard scalarisation methodology (Section 2.2), the RTS method works by reframing the robust multi-objective optimisation problem (1) as a collection of RTS problems. Intuitively, each of these scalarised problems corresponds to one specific preference $\boldsymbol{\theta} \in \Theta$ and one set of solutions that are RTS robust with respect to this preference. By solving a collection of these problems, one hopes to identify a collection of RTS robust solutions that are desirable for the decision maker. As these RTS problems are all scalar-valued, one can in principle appeal to many standard techniques from single-objective optimisation in order to solve them.

The multivariate risk functional completely characterises the notion of robustness in an RTS problem. At a high level, one can interpret the risk-adjusted output set $\boldsymbol{\rho}[f(\mathbf{x},\cdot)] \subseteq \mathbb{R}^M$ as being equivalent to the risk-adjusted Pareto front of an input $\mathbf{x} \in \mathbb{X}$. In the special case where the risk-adjusted set only contains one point, the RTS problem reduces down to a standard scalarised optimisation problem (3), where the vector-valued objective function $g : \mathbb{X} \to \mathbb{R}^M$ is replaced with some risk-penalised alternative. For instance, this occurs when we use any of the component-wise risk functionals (Example 2.11) such as the multivariate expectation (Example 2.8). Note that we can also recover the original scalarised optimisation problems in (3) by letting $\boldsymbol{\rho}[f(\mathbf{x},\cdot)] = \{g(\mathbf{x})\}$ for all $\mathbf{x} \in \mathbb{X}$.

Union robust optimisation problem. Recall from Section 2.2 that it is possible to recover any weakly Pareto optimal solution by solving a collection of length scalarised problems—see Theorem 2.1. We will now present the corresponding analogue of this result for the RTS setting.

Proposition 3.1 (Union robust weak solutions) *Consider a bounded objective function* $f :$ $\mathbb{X} \times \Xi \to \mathbb{R}^M$, *a bounded multivariate risk functional* $\boldsymbol{\rho}$ *and a reference vector* $\boldsymbol{\eta} \in \cap_{\mathbf{x} \in \mathbb{X}} \cap_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]}$ $\mathbb{D}_{\prec}[\{\mathbf{y}\}]$ *which is strongly dominated by the whole risk-feasible objective space. Then, for any weakly union robust input* $\mathbf{z} \in \mathbb{X}$, *there exists a positive unit vector* $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$ *such that the input* $\mathbf{z}$ *is a solution to the RTS problem*

$$\max_{\mathbf{x} \in \mathbb{X}} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]} s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(\mathbf{y}) = \max_{\mathbf{x} \in \mathbb{X}} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]} \min_{m=1,\dots,M} \frac{y^{(m)} - \eta^{(m)}}{\lambda^{(m)}}. \tag{9}$$

To elaborate, Proposition 3.1 shows that the corresponding RTS length scalarised problem (9) can be used in order to target all of the weak solutions to the bounded vector-valued optimisation problem

$$\max\left(\bigcup_{\mathbf{x} \in \mathbb{X}} \boldsymbol{\rho}[f(\mathbf{x},\cdot)] \right), \tag{10}$$

where $\boldsymbol{\rho}$ is a bounded multivariate risk functional. We refer to this problem as the union robust optimisation problem. Conceptually, this optimisation problem treats every input $\mathbf{x} \in \mathbb{X}$ as being equal to its bounded risk-adjusted output set $\boldsymbol{\rho}[f(\mathbf{x},\cdot)] \subset \mathbb{R}^M$. An input $\mathbf{x} \in \mathbb{X}$ will then be considered as being weakly or strictly union robust for this problem if one of its output vectors $\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]$ lies in the overall weak or strict Pareto front of (10), respectively. Through a simple monotonicity argument, one can adapt Theorem 2.1 in order to show that any weak union robust solution can be obtained as a solution to an RTS problem (9). For completeness, we state the corresponding adaptation of this result in Proposition 3.1 and prove it in Appendix A.3.

3.2 Scalarise then robustify

We now introduce the other general class of robust multi-objective optimisation problems which we refer to as the scalarise then robustify problems. A scalar-valued optimisation problem belongs to this class if it can be written in the form

$$\max_{\mathbf{x} \in \mathbb{X}} \rho[s_{\boldsymbol{\theta}}(f(\mathbf{x}, \cdot))], \tag{11}$$

where $s_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}$ denotes a scalarisation function and ρ denotes a univariate risk functional (Section 2.3.1). Conceptually, an STR problem associates each input $\mathbf{x} \in \mathbb{X}$ with its risk-adjusted scalarised value: $\rho[s(f(\mathbf{x}, \cdot))] \in \mathbb{R}$. The solution of an STR problem is then given by the subset of inputs which map to the largest possible risk-adjusted scalarised value. We refer to these solutions as the STR robust solutions of f (with respect to $s_{\boldsymbol{\theta}}$ and ρ). Similar with the RTS method (Section 3.1), the STR method works by reframing the robust multi-objective optimisation problem (1) into a collection of STR problems. As before, each scalarised problem corresponds to one specific preference $\boldsymbol{\theta} \in \Theta$ and one set of solutions that are STR robust with respect to this preference. By solving a collection of these problems, one hopes to identify a collection of STR robust solutions that are desirable for the decision maker. Moreover, as these problems are all scalar-valued, one can in theory appeal to many standard techniques from single-objective optimisation in order to solve them.

The univariate risk functional ρ completely characterises the notion of robustness in an STR problem. It determines the risk-adjusted quality of the bounded output set $f(\mathbf{x}, \Xi) \subset \mathbb{R}^M$ according to the preference encoded by the scalarisation function $s_{\boldsymbol{\theta}}$. Note that we can recover the standard scalarised optimisation problems (3) in the special case when we set $\rho[s_{\boldsymbol{\theta}}(f(\mathbf{x}, \cdot))] = s_{\boldsymbol{\theta}}(g(\mathbf{x}))$ for all $\mathbf{x} \in \mathbb{X}$.

STR length scalarised problem. As with the RTS approach (9), one can also define a collection of STR length scalarised problems of the form

$$\max_{\mathbf{x} \in \mathbb{X}} \rho[s^{\text{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(f(\mathbf{x}, \cdot))] = \max_{\mathbf{x} \in \mathbb{X}} \rho\left[\min_{m=1,\dots,M} \frac{f^{(m)}(\mathbf{x}, \cdot) - \eta^{(m)}}{\lambda^{(m)}}\right], \tag{12}$$

for any partially coherent univariate risk functional ρ, strongly dominated reference vector $\boldsymbol{\eta} \in \cap_{\mathbf{x} \in \mathbb{X}} \cap_{\mathbf{y} \in f(\mathbf{x}, \Xi)} \mathbb{D}_{\prec\!\prec}[\{\mathbf{y}\}]$ and positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. Later on in Section 4, we will show that this collection of length scalarised problems is naturally related with the risk statistics associated with the random Pareto front surface of f. More precisely, we will see that solving the collection of the STR length scalarised problems (12) is in some sense equivalent to optimising for a robust Pareto front surface: namely, the STR front (Definition 4.4). In an analogous way, we will also show that solving the collection of RTS length scalarised problems (9) is in some sense equivalent to optimising for an RTS front (Definition 4.3).

3.3 Discussion

The RTS and STR approach are two general computationally-driven strategies for robust multi-objective optimisation. These two approaches are very flexible and lightweight because they only require the decision maker to specify a sensible family of scalarisation functions and a risk functional in order to define robustness. The former set of functions is determined by the decision maker's internal preferences and specific goals, whilst the latter functional is determined by the uncertainties in the problem and the decision maker's adversity to risk. In this section, we will first discuss the relationships between our methodology and earlier work (Section 3.3.1). We will then look at some natural questions that one might have regarding

these two approaches, such as: which approach should one use in practice (Section 3.3.2) and when are these two approaches equivalent (Section 3.3.3)? Moreover, as an aside, we will also relate some of these ideas with the Gerstewitz functional (Section 3.3.4) and existing concepts from multi-objective reinforcement learning (Section 3.3.5).

3.3.1 Related work

Early work on the topic of robust multi-objective optimisation has focussed on treating the problem (1) as being equivalent to the set-valued optimisation problem [Khan et al., 2015],

$$\max_{\mathbf{x} \in \mathbb{X}} f(\mathbf{x}, \Xi), \tag{13}$$

where the maximum here is defined using some set less relations. That is, an input $\mathbf{x} \in \mathbb{X}$ would be considered robust if there does not exist another distinctive input $\mathbf{x}' \in \mathbb{X} \setminus \{\mathbf{x}\}$ whose output set (7) dominates its own. For example, Ehrgott et al. [2014] proposed using the upper set less relation (Definition 3.1), whilst Ide and Köbis [2014] and Ide et al. [2014] proposed many other alternatives such as the lower set less relation (Definition 3.2). As pointed out in these references, all of these proposed approaches reduce down to the standard Pareto partial ordering over vectors (Definition 2.1) when there is only one possible uncertain parameter $|\Xi| = 1$. An illustration of both of these set-orderings is presented on the left of Figure 3.

Definition 3.1 (Upper set domination) *A set $A \subseteq \mathbb{R}^M$ upper dominates a set $B \subseteq \mathbb{R}^M$ if and only if its dominating region is contained in the weak domination region of the other:* $A \diamond_u B \iff \mathbb{D}_\diamond[A] \subseteq \mathbb{D}_\succeq[B]$, *for* $(\diamond_u, \diamond) \in \{(\succeq_u, \succeq), (\succ_u, \succ), (\succ\!\!\succ_u, \succ\!\!\succ)\}$.

Definition 3.2 (Lower set domination) *A set $A \subseteq \mathbb{R}^M$ lower dominates a set $B \subseteq \mathbb{R}^M$ if and only if its dominated region is contained in the weak dominated region of the other:* $A \diamond_l B \iff \mathbb{D}_\diamond[A] \supseteq \mathbb{D}_\preceq[B]$, *for* $(\diamond_l, \diamond) \in \{(\succeq_l, \preceq), (\succ_l, \prec), (\succ\!\!\succ_l, \prec\!\!\prec)\}$.

In general, we cannot solve the set-valued optimisation problem (13) in its entirety. Instead, most existing optimisation strategies work by identifying some finite approximation to this problem based on some sensible adaptation to the scalarisation approach (Section 2.2). To elaborate, one usually aims to define and solve a finite collection of scalar-valued optimisation problems whose solutions are robust with respect to the set less relation of interest. Below, we will recall some of these examples for the upper and lower set ordered problem. Moreover, we highlight how these examples are just special cases of our RTS and STR approach.

Objective-wise worst-case problem. One mainstream approach to robust single-objective optimisation is the robust counterpart approach [Ben-Tal et al., 2009]. In this approach, one does not treat the output of an objective function by its entire collection of possibilities (7), but instead by some conservative under-estimator of this set: namely, the worst-case value to the output set. The analogue of this approach to the multi-objective setting was presented in earlier work by Kuroiwa and Lee [2012], Fliege and Werner [2014] and Ehrgott et al. [2014]. More specifically, they considered the problem of optimising the objective-wise worst-case optimisation problem

$$\max_{\mathbf{x} \in \mathbb{X}} (\inf_{\boldsymbol{\xi} \in \Xi} f^{(1)}(\mathbf{x}, \boldsymbol{\xi}), \dots, \inf_{\boldsymbol{\xi} \in \Xi} f^{(M)}(\mathbf{x}, \boldsymbol{\xi})). \tag{14}$$

Evidently, this problem is just an instance of the standard vector-valued optimisation problem (2) and therefore it can be solved using the scalarisation approach described earlier in Section 2.2. As explained by Ehrgott et al. [2014], one can obtain solutions to the set-valued optimisation problem (13), under the upper set less relation, by solving different scalarised instances of this worst-case problem (14). Conceptually, this overall computational strategy falls
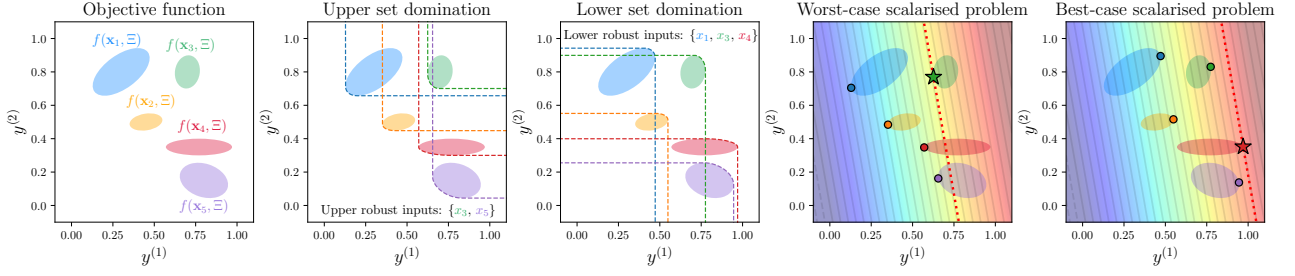
Figure 3: *An illustration of the upper set and lower set ordered optimisation problem* (13) *based on five output sets. On the left, we highlight the corresponding robust points; on the right, we illustrate how one can solve for these points via the scalarisation-based approaches.*

into our class of methods known as the robustify then scalarise approaches. Explicitly, it is the RTS problem where we robustify using the multivariate worst-case functional (Example 2.7) and scalarise using something like the length scalarisation functions (Theorem 2.1).

Extreme case scalarised problems. Alternatively, as suggested in the work by Ehrgott et al. [2014] and Schmidt et al. [2019], we can also identify some upper set robust inputs to (13) by instead solving a collection of worst-case scalarised problems

$$\max_{\mathbf{x} \in \mathbb{X}} \inf_{\boldsymbol{\xi} \in \Xi} s_{\boldsymbol{\theta}}(f(\mathbf{x}, \boldsymbol{\xi})), \tag{15}$$

for some parametric set of scalarisation functions $s_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}$. For example, Ehrgott et al. [2014] proposed solving a collection of linearly scalarised problems[3] which are defined using the linear scalarisation function $s_{\mathbf{w}}^{\mathrm{Lin}}(\mathbf{y}) = \sum_{m=1}^{M} w^{(m)} y^{(m)}$ for $\mathbf{y} \in \mathbb{R}^M$, with different weight vectors $\mathbf{w} \in \Delta^{M-1}$. Whereas Schmidt et al. [2019] proposed solving a collection of Chebyshev scalarised problems, which could equivalently be transformed into a collection of length scalarised problems (5). Note that a similar analysis of this kind was also conducted by Ide and Köbis [2014] for the lower set ordered problem. In this latter case, the scalar-valued problems of interest are the best-case scalarised problems

$$\max_{\mathbf{x} \in \mathbb{X}} \sup_{\boldsymbol{\xi} \in \Xi} s_{\boldsymbol{\theta}}(f(\mathbf{x}, \boldsymbol{\xi})). \tag{16}$$

Conceptually, both of these scalarised problems fall into our class of methods known as the scalarise then robustify approaches. Explicitly, the worst-case (15) and best-case (16) problems scalarises first with some function $s_{\boldsymbol{\theta}}$ and then robustifies using the worst-case and best-case functionals (Example 2.2), respectively. On the right of Figure 3, we showcase how both of these approaches can be used along with the linear scalarisation function in order to identify some of the upper and lower set robust solutions to the set-valued optimisation problem (13).

3.3.2 When to robustify and scalarise?

Computationally, the only difference between the RTS and STR approach is the ordering of the operations. Philosophically though, these two approaches differ in both their goals and their interpretations. In the following paragraphs, we elaborate on these differences in some more detail.

---

[3]Note that a special instance of these robust linearly scalarised problems also appeared in earlier work by Schöttle and Werner [2009] and Fliege and Werner [2014] for the mean-variance portfolio optimisation problem.

Many-time versus one-time philosophy. In general, the RTS approach is suited for problems where the objective function is going to be evaluated many times at the same input and the goal of interest is to maximise the aggregated or summarised performance from these evaluations. In contrast, the STR approach is suited for problems where the objective function will only be evaluated a finite number of times at any given input and the goal is to maximise the consistent one-time performance at this input. Example 3.1 presents a concrete example of this many-time versus one-time philosophy for the target optimisation problem described in Example 2.1.

Example 3.1 (Distance to the ideal vector) *Consider the standard Lp scalarisation function (4) with a uniform weight and the expectation functionals (Examples 2.3 and 2.8). The corresponding RTS problem with this set-up would focus on finding the element that minimises the L^p-distance between expected value and the ideal point:*

$$\min_{\mathbf{x}\in\mathbb{X}} ||\boldsymbol{v} - \mathbb{E}_{p(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})]||_{L^p}.$$

In contrast, the corresponding STR problem would focus on finding the element that minimises the expected L^p-distance between the objective function and the ideal point:

$$\min_{\mathbf{x}\in\mathbb{X}} \mathbb{E}_{p(\boldsymbol{\xi})}[||\boldsymbol{v} - f(\mathbf{x},\boldsymbol{\xi})||_{L^p}].$$

Evidently, these two formulations are different. The former approach is intended for the many-time use setting, where one is interested in identifying an input $\mathbf{x} \in \mathbb{X}$ whose aggregated performance is close to the ideal point. In contrast, the latter approach is designed for the one-time use setting, where one is interested in identifying the input $\mathbf{x} \in \mathbb{X}$ whose sample output vectors are consistently close to the ideal point.

Interpretability. In practice, decision makers are interested in making principled decisions that can be justified and explained to other stakeholders. From an interpretation perspective, the RTS approach is appealing because it can easily be interpreted as a standard multi-objective optimisation problem (2) wherein one replaces the objective function with a risk-penalised version (10). The corresponding scalarisation methodology can then be viewed as an operational technique in order to acquire the robust points which are the most desirable. In contrast, the STR approach takes a much more preference-driven strategy that is based largely on the scalarisation methodology. Specifically, one has to first identify a scalarisation function that describes the most preferred total ordering before one can define robustness. This is clearly advantageous in the settings where a preference exists on the outset. But in general, as with the RTS approach, one typically has to keep track of multiple preferences at the beginning before refining them later on. Alternatively, as described earlier in Section 3.3.1, one can also view the STR approach as a computational strategy that is posed in order to solve some set-valued optimisation problem (13).

3.3.3 Commutative operations

There are some special cases where the RTS problem and STR problem are equivalent in the sense that the maximisers are the same, that is

$$\arg\max_{\mathbf{x}\in\mathbb{X}} \max_{\mathbf{y}\in\boldsymbol{\rho}[f(\mathbf{x},\cdot)]} s(\mathbf{y}) = \arg\max_{\mathbf{x}\in\mathbb{X}} \rho[s'(f(\mathbf{x},\cdot))]$$

for some potentially different scalarisation functions s, s' and risk functionals $\rho, \boldsymbol{\rho}$. For example, as described in the previous sections, we can recover the original scalarised optimisation problem

([3](#)) as a special case of both the RTS and STR problems. In the following, we present some additional examples where the robustification and scalarisation operation commutes in the above sense. Notably, the primary benefit of this commutativity property is that it gives us a way to get the best of both worlds. That is, we can selectively inherit any desirable features from either the RTS and STR approach in order to justify and solve the robust problem of interest.

Example 3.2 (Best-case scalarised value) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and a scalarisation function $s : \mathbb{R}^M \to \mathbb{R}$. By simply expanding the definitions, we see that the RTS problem with the identity functional (Example [2.6](#)) and the STR problem with the best-case functional (Example [2.2](#)) with $\mathcal{E} = \Xi$ are equivalent:*

$$\max_{\mathbf{y} \in f(\mathbf{x}, \Xi)} s(\mathbf{y}) = \max_{\boldsymbol{\xi} \in \Xi} s(f(\mathbf{x}, \boldsymbol{\xi}))$$

for any input $\mathbf{x} \in \mathbb{X}$.

Example 3.3 (Worst-case length scalarised value) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, the length scalarisation function ([5](#)) and the worst-case functionals (Examples [2.2](#) and [2.7](#)). By simply expanding the definitions and switching the order of the minimisation and infimum operations, we see that the corresponding RTS and STR problem with this set-up are equivalent:*

$$s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}\left(\left(\inf_{\boldsymbol{\xi} \in \mathcal{E}} f^{(1)}(\mathbf{x}, \boldsymbol{\xi}), \ldots, \inf_{\boldsymbol{\xi} \in \mathcal{E}} f^{(M)}(\mathbf{x}, \boldsymbol{\xi})\right)\right) = \inf_{\boldsymbol{\xi} \in \mathcal{E}} s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(f(\mathbf{x}, \boldsymbol{\xi}))$$

for any uncertainty set $\mathcal{E} \subseteq \Xi$, reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$ and input $\mathbf{x} \in \mathbb{X}$.

Example 3.4 (Linearity of the expectation) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, the linear scalarisation function and the expectation functionals (Examples [2.3](#) and [2.8](#)). By the linearity of the expectation, the corresponding RTS and STR problem with this set-up are equivalent:*

$$s^{\mathrm{Lin}}_{\mathbf{w}}(\mathbb{E}_{p(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]) = \mathbb{E}_{p(\boldsymbol{\xi})}[s^{\mathrm{Lin}}_{\mathbf{w}}(f(\mathbf{x}, \boldsymbol{\xi}))]$$

for any distribution $p \in \mathcal{P}[\Xi]$, weight vector $\mathbf{w} \in \Delta^{M-1}$ and input $\mathbf{x} \in \mathbb{X}$.

Example 3.5 (Value-at-risk of the lengths) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, the length scalarisation function ([5](#)) and the VaR functionals (Examples [2.4](#) and [2.9](#)). By a standard property of the Chebyshev scalarisation function [Daulton et al., 2022, Theorem 5.1], the corresponding RTS and STR problem with this set-up are equivalent in the following sense:*

$$\max_{\mathbf{y} \in \boldsymbol{\rho}^{\mathrm{VaR}_\alpha}[f(\mathbf{x}, \cdot)]} s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(\mathbf{y}) = \rho^{\mathrm{VaR}_\alpha}[s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(f(\mathbf{x}, \cdot))]$$

for any distribution $p \in \mathcal{P}[\Xi]$, level $\alpha \in (0, 1)$, strongly dominated reference vector $\boldsymbol{\eta} \in \bigcap_{(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{X} \times \Xi} \mathbb{D}_{\prec\!\prec}[\{f(\mathbf{x}, \boldsymbol{\xi})\}]$, positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$ and input $\mathbf{x} \in \mathbb{X}$.

3.3.4 General nonlinear scalarisation

As showcased in the works by Klamroth et al. [2013, 2017], one can unify many different concepts from robust optimisation and stochastic programming via the general class of nonlinear scalarising functionals introduced by Gerstewitz (later Gerth, now Tammer) [Gerstewitz, 1983, Tammer and Weidner, 2020]. For a topological vector space V, the Gerstewitz functional $\varphi_{r-C,k} : V \to \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ is defined by

$$\varphi_{r-C,k}(v) = \inf\{t \in \mathbb{R} : v \in r - C + tk\}, \tag{17}$$

where C is a proper closed subset of V, $r \in V$ is a reference point, $k \in V \setminus \{0\}$ is a direction of improvement and $C + k\mathbb{R}_{\geq 0} \subseteq C$. As highlighted in these works, many well-known scalarised minimisation problems are equivalent to minimising an instance of $\varphi_{r-C,k}$ over some subset $W \subseteq V$, that is

$$\min_{v \in W} \varphi_{r-C,k}(v). \tag{18}$$

For example, in the context of multi-objective optimisation, when $V = \mathbb{R}^M$, we can recover the Chebyshev, ϵ-constraint [Ehrgott, 2005, Section 4.1] and Pascoletti-Serafini [Pascoletti and Serafini, 1984] scalarised problems as special cases. Similarly, in the context of robust or stochastic single-objective optimisation, when $V = \mathbb{R}^\Xi$, we can recover the worst-case and expectation optimisation problems as special cases. Notably, in the context of both the RTS and STR approaches, we can naturally substitute instances of this functional in place of either the scalarisation function or risk functional (or both).

Length scalarisation function. As the length scalarisation function (5) is just a reparameterisation of the Chebyshev scalarisation function, it can also be recovered as a special case of (18). More specifically, when $V = \mathbb{R}^M$ we have that

$$\max_{\mathbf{y} \in Y} s^{\text{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(\mathbf{y}) = \max_{\mathbf{y} \in Y} \sup\{t \in \mathbb{R} : \boldsymbol{\eta} + t\boldsymbol{\lambda} \in \mathbf{y} - \mathbb{R}^M_{\geq 0}\}$$

$$= -\min_{\mathbf{y} \in Y} \inf\{t \in \mathbb{R} : -\mathbf{y} \in -\boldsymbol{\eta} - \mathbb{R}^M_{\geq 0} + t\boldsymbol{\lambda}\}$$

$$= -\min_{\mathbf{y} \in Y} \varphi_{-\boldsymbol{\eta} - \mathbb{R}^M_{\geq 0}, \boldsymbol{\lambda}}(-\mathbf{y})$$

where $Y \subseteq \mathbb{D}_{\succcurlyeq}[\{\boldsymbol{\eta}\}]$. It is unclear to us whether the RTS and STR variant of this problem, (9) and (12), can also be written as special instances of (18).

Coherent risk measures. In the context of single-objective optimisation, Quaranta and Zaffaroni [2008] established links between robustification of linear programs and coherent risk measures such as the CVaR. In this work, we only use some of the properties of a coherent risk measure (Definition 2.4) in order to establish the Pareto front risk statistics (Proposition 4.1) and the STR front (Proposition 4.3). In general, outside of these specific results, our framework allows one to robustify with risk measures that are not necessarily coherent such as the VaR (Example 2.4).

As explained by Artzner et al. [1999, Definition 2.2], Klamroth et al. [2013, Section 6] and Tammer and Weidner [2020, Section 15.1.1], coherent risk measures (Example 2.5) can also be written as Gerstewitz functionals (17). Generalising our work in terms of these functionals and better understanding its deep connections with coherence and risk are beyond the scope of this paper. These are however very interesting directions for future work.

3.3.5 Connections with reinforcement learning

Another instance of this RTS versus STR problem has appeared earlier in the topic of multi-objective reinforcement learning [Roijers et al., 2013]. In this class of problems, one is interested in identifying policies that maximise the expected cumulative reward associated with a vector-valued Markov decision process. As motivated by Roijers et al. [2013], scalarisation functions play a prominent role in this problem because they decide which policies one should select and then execute in practice. As highlighted by them, there are two possible scalarised objectives one could define in practice: the scalarised expected return (SER) or the expected scalarised return (ESR). In essence, the SER and ESR approach are just special instances of the RTS and STR approach, respectively. Explicitly, these are the instances where we use the expectation

as our risk functional (Examples 2.3 and 2.8). The philosophical debate on whether one should take an SER or ESR approach is also similar to the RTS versus STR problem (Section 3.3.2). Specifically, Roijers et al. [2013, Section 8.4] says that one should use an SER approach when *"the policy will be used many times and return accumulates across episodes"*. Whereas, an ESR approach should be used when *"the policy will only be used a few times or the return does not accumulate across episodes"*. Note however that in the context of reinforcement learning, the notion of accumulation is that the rewards are summed or averaged. In contrast, in our set-up, we consider the more general notion of aggregation, where the rewards are not necessarily summed but just aggregated into a single or set of summary values via a risk functional.

4   Robust Pareto fronts

The Pareto front is an important quantity in standard multi-objective optimisation (2). It gives us valuable information about the different trade-offs that are happening in the objective space. In practice, this knowledge is typically used by a decision maker in order to help them make an informed and effective decision. So far, much of the work on robust multi-objective optimisation has focussed on classifying and identifying inputs that are deemed as robust in some sense. To the best of our knowledge no work has focussed on identifying and interpreting the robust trade-off that is happening in the objective space. This section addresses this gap in the literature by proposing a principled way to define a robust Pareto front under the RTS and STR perspectives for the length scalarised problems. More precisely, our work leverages ideas from a recent paper by Tu et al. [2024], who showed how it is possible to reconstruct an interpolated Pareto front surface from a collection of length scalarised values. This key result was referred to as the polar parameterisation of a Pareto front surface, which we first recall in Section 4.1. Then we move on to Section 4.2 where we exploit this parameterisation result in order to define the general concept of a risk statistic associated with a random Pareto front surface (Proposition 4.1). Afterwards, in Section 4.3, we present the novel concept of an RTS front (Definition 4.3) and a STR front (Definition 4.4). Formally, these are the surfaces described by the corresponding RTS and STR length scalarised values, respectively. As we prove in the main results in this section, Proposition 4.2 and Proposition 4.3, both of these surfaces are Pareto front surfaces, under mild assumptions, and therefore are valid robust generalisations to the standard Pareto front trade-off surface. Finally, as an aside, in we present a theoretical result in Section 4.4 regarding the ordering of the extreme case Pareto front surfaces.

4.1   Polar parameterisation

To begin with, we recall the definition of a Pareto front surface. Intuitively, this surface is the one obtained by interpolating the strict Pareto front of a set using the weak Pareto partial ordering. As this interpolation could extend indefinitely, we consider truncating this surface at some lower bound defined by a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. Afterwards, we recall the definition of a polar surface.

Definition 4.1 (Pareto front surface) *The Pareto front surface of a bounded set of vectors $A \subset \mathbb{R}^M$, with respect to a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, is defined as the truncated weak Pareto front of its weak domination after closure, that is $\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A] := \mathcal{Y}^{\mathrm{weak}}[\mathbb{D}_{\preceq}[\mathrm{CLOSURE}(A)]] \cap \mathbb{D}_{\succcurlyeq}[\{\boldsymbol{\eta}\}]$. The set of all non-empty Pareto front surfaces with respect to the reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$ is denoted by $\mathbb{Y}_{\boldsymbol{\eta}}^* \subset 2^{\mathbb{R}^M}$.*

Definition 4.2 (Polar surface) *A set $A \subseteq \mathbb{R}^M$ is called a polar surface, with respect to a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, if and only if there exists a unique non-negative bounded function*

17

$r_{\boldsymbol{\eta},A} : \mathcal{S}_+^{M-1} \to \mathbb{R}_{\geq 0}$ *such that*

$$A = \{\boldsymbol{\eta} + r_{\boldsymbol{\eta},A}(\boldsymbol{\lambda})\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}\}.$$

The set of all polar surfaces with respect to a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$ is denoted by $\mathbb{L}_{\boldsymbol{\eta}} \subset 2^{\mathbb{R}^M}$. Moreover, for this set $\mathbb{L}_{\boldsymbol{\eta}}$, we define the projected length function $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}} : \mathbb{L}_{\boldsymbol{\eta}} \to \mathbb{R}_{\geq 0}$ to be the function that returns the projected length along any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$, that is $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[A] = r_{\boldsymbol{\eta},A}(\boldsymbol{\lambda})$ for any $A \in \mathbb{L}_{\boldsymbol{\eta}}$.

Geometrically, a polar surface is any set that can be obtained by stretching the positive space of unit vectors $\mathcal{S}_+^{M-1}$ along the radial directions, before translating the origin to the reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. Consequently, this implies that a polar surface can be described entirely by its projected length function $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}$, which computes the amount of stretching that occurs along any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. As shown by Tu et al. [2024], and recalled in Theorem 4.1 below, every Pareto front surface is a polar surface. In particular, it is a polar surface whose projected length function can be written in terms of the length scalarisation function (5). On the left of Figure 4, we present a pictorial illustration of this parameterisation result for the Pareto front surface described in Figure 1.

Theorem 4.1 (Polar parameterisation) *[Tu et al., 2024, Theorem 3.1] For any bounded set of vectors $A \subset \mathbb{R}^M$ and reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, if the corresponding Pareto front surface is non-empty $\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A] \neq \emptyset$, then it admits the following polar parameterisation:*

$$\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A] = \left\{ \boldsymbol{\eta} + \sup_{\mathbf{a} \in A} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(\mathbf{a})\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}_+^{M-1} \right\} \tag{19}$$

where $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A]] = \sup_{\mathbf{a} \in A} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(\mathbf{a})$ is the projected length of A along the positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$.

Remark 4.1 (Singleton front) *When the Pareto front surface is empty, $\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A] = \emptyset$, then the right hand side of (19) evaluates to the degenerate polar surface $\{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$. This event can happen when the reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$ is set too aggressively in such a way as it weakly dominates the entire feasible objective space.*

4.2 Risk statistics

To apply the polar parameterisation result on the objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, with a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, we need to ensure that it fulfils the necessary assumptions outlined in Theorem 4.1. That is, we need to ensure that the objective function is bounded and that the corresponding Pareto front surface is non-empty. In this work, we only assume that the objective function is bounded. As a result, the polar parameterisation of any bounded set of points could potentially degenerate. Note that this event can however be avoided if the reference vector is set more judiciously (Remark 4.1). Formally, in this work, we consider the family of polar surfaces

$$Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi}) := \left\{ \boldsymbol{\eta} + \sup_{\mathbf{x} \in \mathbb{X}} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(f(\mathbf{x},\boldsymbol{\xi}))\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}_+^{M-1} \right\}, \tag{20}$$

with $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi})] = \sup_{\mathbf{x} \in \mathbb{X}} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(f(\mathbf{x},\boldsymbol{\xi}))$, for any uncertain vector $\boldsymbol{\xi} \in \Xi$ and positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. If the objective function f is bounded, then this set is either equal to the Pareto front surface $\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[\{f(\mathbf{x},\boldsymbol{\xi})\}_{\mathbf{x} \in \mathbb{X}}]$, if it is non-empty, or it is the degenerate singleton set $\{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$. As shown by Tu et al. [2024, Section 4], one can readily use this polar representation in order to generalise standard statistics from the univariate setting to the space of Pareto front surfaces. For example, one can define the expected Pareto front surface as the polar surface constructed using the expected projected lengths (Example 4.1).
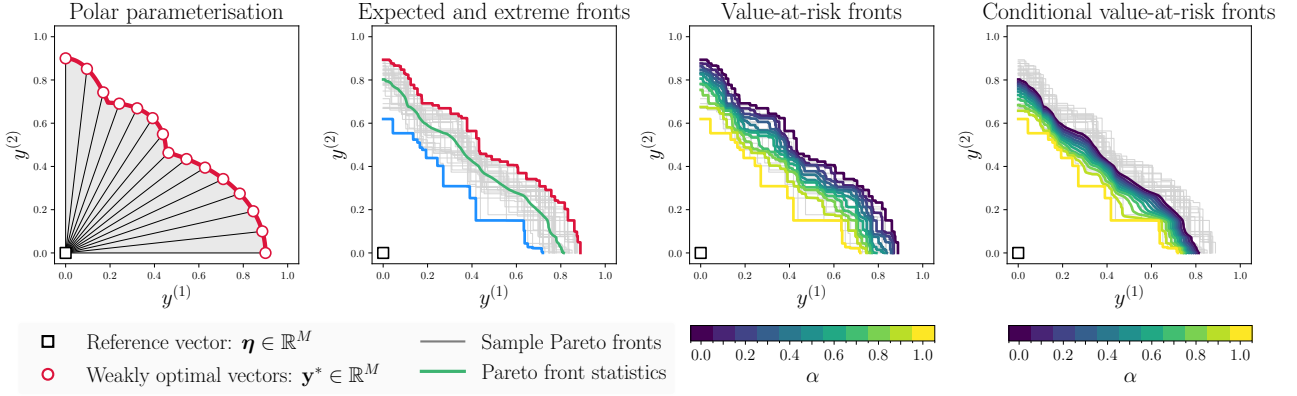
Figure 4: *We illustrate the polar parameterisation (Theorem 4.1) and the Pareto front surface statistics (Proposition 4.1) for a simple two-dimensional example. On the left, we plot the polar parameterisation of the Pareto front surface from Figure 1. On the remaining plots, we draw the corresponding risk statistics associated with a finite collection of Pareto front surfaces and the partially coherent univariate risk functionals listed in Section 2.3.1.*

Example 4.1 (Expected Pareto front) *The expectation of the set* (20) *under some distribution $p \in \mathcal{P}[\Xi]$ is given by the polar surface*

$$\mathbb{E}_{p(\boldsymbol{\xi})}[Y^*_{\boldsymbol{\eta},f}(\boldsymbol{\xi})] := \{\boldsymbol{\eta} + \mathbb{E}_{p(\boldsymbol{\xi})}[\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y^*_{\boldsymbol{\eta},f}(\boldsymbol{\xi})]]\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+\}.$$

As shown by Tu et al. [2024, Section 4.1], if the objective function f is bounded and the Pareto front surfaces $\mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[\{f(\mathbf{x}, \boldsymbol{\xi})\}_{\mathbf{x} \in \mathbb{X}}]$ are non-empty almost surely, then this polar surface is a valid Pareto front surface. Otherwise it could degenerate to the singleton set $\{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

We now present a straightforward extension of this result below, which holds for any partially coherent univariate risk functional. The proof of this result is presented in Appendix A.6.

Proposition 4.1 (Pareto front surface statistics) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. Then, for any partially coherent univariate risk functional ρ, the set*

$$\rho[Y^*_{\boldsymbol{\eta},f}] := \{\boldsymbol{\eta} + \rho[\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y^*_{\boldsymbol{\eta},f}(\cdot)]]\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+\} \tag{21}$$

is either a Pareto front surface $\rho[Y^*_{\boldsymbol{\eta},f}] \in \mathbb{Y}^*_{\boldsymbol{\eta}}$ or the singleton set $\{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

On the right of Figure 4, we illustrate a number of different risk statistics (21) for a two-dimensional Pareto front surface distribution. In fact, we plot the corresponding empirical estimate for the Pareto surface statistics based on all of the partially coherent univariate risk functionals that were listed in Section 2.3.1.

Remark 4.2 (Optimising a risk statistic) *Similar to the RTS and STR approach, one could also define the robust optimisation problem* (1) *as being equal to the problem of optimising a Pareto front surface statistic* (21). *That is, we aim to solve a collection of set optimisation problems of the form*

$$\max_{X \subseteq \mathbb{X}, |X| \leq P} \rho\left[\max_{\mathbf{x} \in X} s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(f(\mathbf{x}, \cdot))\right],$$

for some positive directions $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$, where $P > 0$ denotes a cardinality constraint. Individually, each of these problems tries to identify a collection of inputs $X \subseteq \mathbb{X}$, whose collective output, under the preference direction $\boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+$, is of high quality. This problem reduces down to the STR length scalarised problem (12) *when $P = 1$. Notably, this is a monotone and submodular set optimisation problem when $P > 1$ and therefore is, in general, much more computationally challenging to solve than the corresponding RTS or STR problems—see Tu et al. [2025] for more details.*

19

4.3 Robust Pareto front surfaces

The polar parameterisation result in Theorem 4.1 tells us that we can reconstruct a Pareto front surface from its corresponding collection of maximum length scalarised values. We will now take advantage of this result in order to define the corresponding polar surfaces for the RTS and STR length scalarised problems. More specifically, we define the RTS front and STR front as the polar surface obtained using the RTS length values (9) and the STR length values (12), respectively.

Definition 4.3 (RTS front) *The RTS front associated with a subset of inputs $X \subseteq \mathbb{X}$, a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and a bounded multivariate risk functional $\boldsymbol{\rho}$ is defined by the set*

$$\mathcal{F}^{\mathrm{RTS}}_{\boldsymbol{\eta}, f, \boldsymbol{\rho}}[X] := \left\{ \boldsymbol{\eta} + \sup_{\mathbf{x} \in X} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]} s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(\mathbf{y}) \boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+ \right\}. \tag{22}$$

Definition 4.4 (STR front) *The STR front associated with a subset of inputs $X \subseteq \mathbb{X}$, a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$, a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and a partially coherent univariate risk functional ρ is defined by the set*

$$\mathcal{F}^{\mathrm{STR}}_{\boldsymbol{\eta}, f, \rho}[X] := \left\{ \boldsymbol{\eta} + \sup_{\mathbf{x} \in X} \rho[s^{\mathrm{Len}}_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}(f(\mathbf{x}, \cdot))] \boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}^{M-1}_+ \right\}. \tag{23}$$

In Proposition 4.2 and Proposition 4.3, we present the main results in this section which states that these polar surfaces (22) and (23) are indeed valid Pareto front surfaces. Also, we give an explicit representation on what these Pareto front surfaces actually are. In words, the RTS front is the Pareto front surface associated with the union robust problem (10), whilst the STR front is the Pareto front surface associated with the risk statistic of each input's Pareto front surface distribution. The proof of these result are presented in Appendix A.7 and Appendix A.8, respectively.

Proposition 4.2 (RTS front) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. Then, for any bounded multivariate risk functional $\boldsymbol{\rho}$ and any subset of inputs $X \subseteq \mathbb{X}$, the corresponding RTS front is either a Pareto front surface*

$$\mathcal{F}^{\mathrm{RTS}}_{\boldsymbol{\eta}, f, \boldsymbol{\rho}}[X] = \mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[\cup_{\mathbf{x} \in X} \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]] = \mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[\cup_{\mathbf{x} \in X} \mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[\boldsymbol{\rho}[f(\mathbf{x}, \cdot)]]] \in \mathbb{Y}^*_{\boldsymbol{\eta}},$$

or the singleton set $\mathcal{F}^{\mathrm{RTS}}_{\boldsymbol{\eta}, f, \boldsymbol{\rho}}[X] = \{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

**Proposition 4.3 (STR front)** *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. Then, for any partially coherent univariate risk functional ρ and any subset of inputs $X \subseteq \mathbb{X}$, the corresponding STR front is either a Pareto front surface*

$$\mathcal{F}^{\mathrm{STR}}_{\boldsymbol{\eta}, f, \rho}[X] = \mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[\cup_{\mathbf{x} \in X} \rho[\mathcal{Y}^{\mathrm{int}}_{\boldsymbol{\eta}}[f(\mathbf{x}, \cdot)]]] \in \mathbb{Y}^*_{\boldsymbol{\eta}},$$

or the singleton set $\mathcal{F}^{\mathrm{STR}}_{\boldsymbol{\eta}, f, \rho}[X] = \{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

In Figure 5, we illustrate these risk-adjusted Pareto front surfaces for a simple two-dimensional example with the risk functionals introduced in Section 2.3. We also compare these Pareto front surfaces with the risk statistics described in Proposition 4.1. Overall, we see a great variety between the Pareto front surfaces illustrated within each plot; the differences among the plots however are very subtle. Conceptually, the Pareto front surface statistics looks at the aggregated performance at each scalarised value and therefore has a noticeably smoother
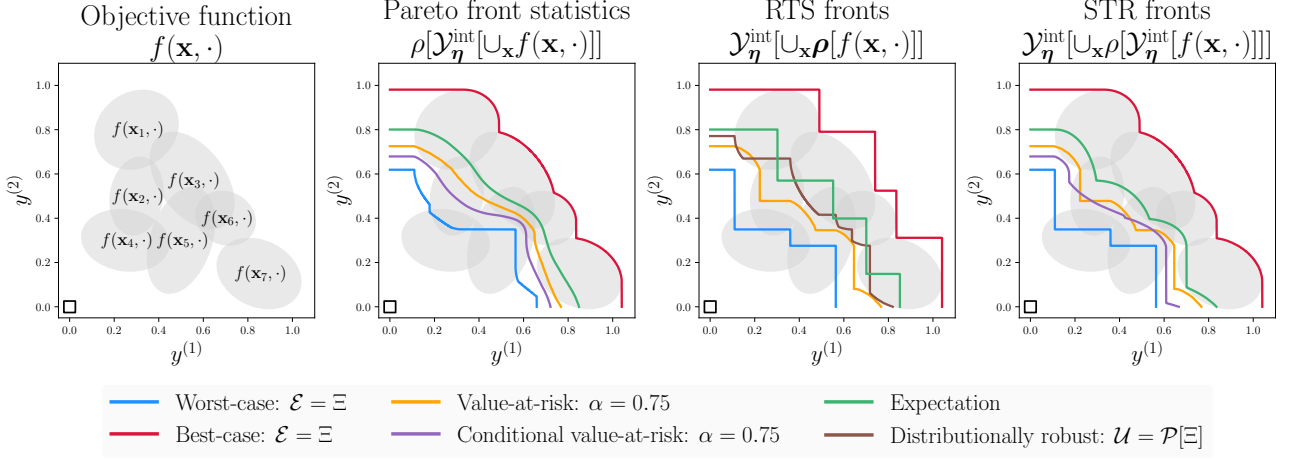
Figure 5: *An illustration of the different Pareto front surfaces that can be computed from an $M = 2$ dimensional objective function. On the left, we plot the output sets $f(\mathbf{x}, \Xi) \subset \mathbb{R}^M$ for each input $\mathbf{x} \in \mathbb{X}$, which are all assumed to be uniformly distributed in some ellipsoid. On the remaining plots, we draw the Pareto front surfaces associated with the Pareto front statistics, RTS fronts and STR fronts for the different risk functionals described in Section 2.3.*

surface. In contrast, the RTS and STR front looks at the individual performance at each scalarised level and therefore is a bit more bumpy. In particular, the RTS front treats every input $\mathbf{x} \in X$ according to the Pareto front surface of its risk-adjusted output set $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\boldsymbol{\rho}[f(\mathbf{x}, \cdot)]]$, whilst the STR front treats every input $\mathbf{x} \in X$ according to a risk statistic of its random Pareto front surface $\boldsymbol{\rho}[\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[f(\mathbf{x}, \cdot)]]$. These two formulations are different in general. However, as highlighted in Section 3.3.3, there are special cases when they are equivalent, such as: the worst-case setting (blue lines) and the VaR setting (orange lines).

4.4 Bounding a nominal surface

In some practical settings, the uncertain parameter $\boldsymbol{\xi} \in \Xi$ denotes some nuisance parameter that obscures the true value of the objective function. That is, we suppose that the true value of the objective function appears when this parameter realises its nominal value of $\boldsymbol{\xi}^* \in \Xi$. Under this set-up, the traditional goal for a practitioner would be to identify the Pareto front surface at this nominal value: $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\{f(\mathbf{x}, \boldsymbol{\xi}^*)\}_{\mathbf{x} \in \mathbb{X}}]$. In general, one cannot do this exactly. Instead, one typically tries to identify a bound for this surface which holds with high probability. In this section, we analyse the gap between the upper and lower bounding surface obtained using the extreme case functionals (Examples 2.2 and 2.7). Our main result in this section is Proposition 4.4. This result states that the gap between the best-case and worst-case Pareto front surfaces, for the different robust variants, are ordered. The smallest gap is attained by the Pareto front surface statistics, followed by the STR fronts and then the RTS fronts. This result is illustrated in Figure 5 using the best-case fronts (red lines) and worst-case fronts (blue lines). We will now set up the notation that is required in order to state this result.

For any bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and strongly dominated reference vector $\boldsymbol{\eta} \in \bigcap_{\mathbf{x} \in \mathbb{X}} \bigcap_{\mathbf{y} \in f(\mathbf{x}, \Xi)} \mathbb{D}_{\prec}[\{\mathbf{y}\}]$, we can write the target Pareto front surface in its polar parameterised form (20). By definition, this implies that the projected lengths of this Pareto front surface $Y_{\boldsymbol{\eta}, f}^*(\boldsymbol{\xi}^*)$, along any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$, can be written as

$$\ell_{\boldsymbol{\eta}, \boldsymbol{\lambda}}[Y_{\boldsymbol{\eta}, f}^*(\boldsymbol{\xi}^*)] = \sup_{\mathbf{x} \in \mathbb{X}} \min_{m=1, \ldots, M} z_{\boldsymbol{\eta}, \boldsymbol{\lambda}}^{(m)}(f^{(m)}(\mathbf{x}, \boldsymbol{\xi}^*)),$$

where the term $z_{\boldsymbol{\eta}, \boldsymbol{\lambda}}^{(m)}(y) = (y - \eta^{(m)})/\lambda^{(m)}$ is used to denote the m-th argument in the length scalarisation function (5) for any number $y \in \mathbb{R}$. It can be easily shown that this Pareto front

21

surface can be bounded from above and below by the extreme cases of the Pareto front surface statistics (denoted by PS), RTS fronts and STR fronts. Mathematically, this statement can be formalised by the following three inequalities:

$$\underbrace{\inf_{\boldsymbol{\xi}\in\Xi}\sup_{\mathbf{x}\in\mathbb{X}}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{PS}}} \leq \ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi}^*)] \leq \underbrace{\sup_{\boldsymbol{\xi}\in\Xi}\sup_{\mathbf{x}\in\mathbb{X}}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{PS}}},$$

$$\underbrace{\sup_{\mathbf{x}\in\mathbb{X}}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(\inf_{\boldsymbol{\xi}\in\Xi}f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{RTS}}} \leq \ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi}^*)] \leq \underbrace{\sup_{\mathbf{x}\in\mathbb{X}}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(\sup_{\boldsymbol{\xi}\in\Xi}f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{RTS}}},$$

$$\underbrace{\sup_{\mathbf{x}\in\mathbb{X}}\inf_{\boldsymbol{\xi}\in\Xi}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{STR}}} \leq \ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi}^*)] \leq \underbrace{\sup_{\mathbf{x}\in\mathbb{X}}\sup_{\boldsymbol{\xi}\in\Xi}\min_{m=1,\ldots,M}z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(f^{(m)}(\mathbf{x},\boldsymbol{\xi}))}_{u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{STR}}},$$

which all hold simultaneously for every uncertain parameter $\boldsymbol{\xi}^* \in \Xi$. We see clearly that the only difference between these bounds are the ordering of the three main operations: maximisation over the inputs, robustification over the uncertain parameters and scalarisation of the objective.

Proposition 4.4 (Extreme case bounds) *For any bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and strongly dominated reference vector $\boldsymbol{\eta} \in \cap_{\mathbf{x}\in\mathbb{X}}\cap_{\mathbf{y}\in f(\mathbf{x},\Xi)}\mathbb{D}_{\prec\!\prec}[\{\mathbf{y}\}]$, the following inequality holds:*
$$0 \leq u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{PS}} - l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{PS}} \leq u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{STR}} - l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{STR}} \leq u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{RTS}} - l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\mathrm{RTS}},$$
for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$.

Proposition 4.4 states that the ordering of these three operations can be exploited in order to get a smaller bounding interval. The proof of this result is presented in Appendix A.9. Note that in the special case when the set of uncertain parameters is a singleton set $\Xi = \{\boldsymbol{\xi}^*\}$, all of these extreme case Pareto front surfaces are equal. In other words, the difference between these three approaches only begins to transpire when the uncertainty set grows. Proposition 4.4 gives us an ordering on the range of Pareto front surfaces one can obtain. It does not however give us any information regarding which method is the most effective to identify or estimate this uncertain parameter $\boldsymbol{\xi}^* \in \Xi$. This latter problem turns out to be case specific as it relies heavily on the assumptions that are placed on the objective function and the form of the uncertainty.

5 Robust performance metrics

Performance metrics are often used in multi-objective optimisation in order to quantify the quality of a Pareto front approximation. As showcased in a recent survey by Audet et al. [2021], there exists many different types of performance metrics in the multi-objective literature, all of which try to measure different facets of the Pareto front approximation. Very little work has focussed on extending these existing performance metrics to the robust setting [Mirjalili and Lewis, 2015]. More specifically, in this latter setting, one wishes to quantify the quality of a set of proposed inputs $X \subseteq \mathbb{X}$, based on the corresponding set of output sets $\{f(\mathbf{x},\Xi)\}_{\mathbf{x}\in X}$. Naturally, if the individual output sets are singletons, $|f(\mathbf{x},\Xi)| = 1$, then we reduce down to the standard multi-objective set-up. In this section, we will now address this gap in the literature, and present two family of robust multi-objective performance metrics based on the RTS and STR approaches. Notably, we generalise the R2 utilities [Hansen and Jaszkiewicz, 1998], which are a general family of multi-objective performance metrics that exhibit many favourable properties and contains many of the most widely-used performance metrics as special cases [Tu et al.,

2025]. As explicit examples, we will proceed to show how one can generalise the well-known inverted generational distance (Section 5.2) and hypervolume indicator (Section 5.3) to the robust setting.

5.1 Robust R2 utilities

Formally, a multi-objective performance metric is any utility function $U : 2^{\mathbb{R}^M} \to \mathbb{R}$, which maps a set of vectors to a scalar. We adopt the convention that larger utility values signify better performance. The R2 utilities are a special class of utility functions that is defined with respect to a family of scalarisation functions $\{s_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}\}_{\boldsymbol{\theta} \in \Theta}$ and a probability distribution over the corresponding scalarisation parameters $p \in \mathcal{P}[\Theta]$. Specifically, equipped with this set-up, the corresponding R2 utility of a set of vectors $Y \subset \mathbb{R}^M$ is equal to the weighted average maximum scalarised value

$$U[Y] := \mathbb{E}_{p(\boldsymbol{\theta})}\left[\sup_{\mathbf{y} \in Y} s_{\boldsymbol{\theta}}(\mathbf{y})\right]. \tag{24}$$

Note that this is the general form of an R2 utility which also accounts for the setting where the Pareto front approximation $Y \subset \mathbb{R}^M$ is continuous. Nevertheless, in practice, the Pareto front approximation is typically finite and therefore the supremum operation is often replaced with the maximum. At an intuitive level, an R2 utility just computes a weighted score associated with a collection of scalarised problems (3). By design, this family of utility functions is very flexible and it gives us any easy way to define and measure different notions of performance in the objective space. In theory, one could take advantage of this flexibility in order to define R2 utilities which adequately cater for the decision maker's internal preferences.

Robust R2 Utility. Inspired by the construction of the R2 utilities (24), we now define the corresponding RTS and STR extension of the R2 utilities by replacing the maximum scalarised values arising in (3) with the RTS scalarised values (8) and STR scalarised values (11), respectively. More precisely, we define an RTS-R2 utility function $R_{f,\boldsymbol{\rho}}^{\mathrm{RTS}} : 2^{\mathbb{X}} \to \mathbb{R}$ and a STR-R2 utility function $R_{f,\rho}^{\mathrm{STR}} : 2^{\mathbb{X}} \to \mathbb{R}$ by the following equations:

$$R_{f,\boldsymbol{\rho}}^{\mathrm{RTS}}[X] := \mathbb{E}_{p(\boldsymbol{\theta})}\left[\sup_{\mathbf{x} \in X} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]} s_{\boldsymbol{\theta}}(\mathbf{y})\right], \tag{25}$$

$$R_{f,\rho}^{\mathrm{STR}}[X] := \mathbb{E}_{p(\boldsymbol{\theta})}\left[\sup_{\mathbf{x} \in X} \rho[s_{\boldsymbol{\theta}}(f(\mathbf{x},\cdot))]\right], \tag{26}$$

for any set of inputs $X \subseteq \mathbb{X}$. Note that in contrast to the original R2 utilities in (24), these robust R2 utilities assess the quality of a set of inputs instead of a set of outputs. Formally, these robust performance metrics are defined with respect to an objective function and a risk functional. This construction makes sense from an intuitive perspective because the output of any input is subject to uncertainty and therefore one has to account for the uncertainty in some way in order to assess the overall robust performance.

Remark 5.1 (RTS-R2 utility) *Notice that one can rewrite an RTS-R2 utility in terms of a standard R2 utility:* $R_{f,\boldsymbol{\rho}}^{\mathrm{RTS}}[X] = U[\cup_{\mathbf{x} \in X} \boldsymbol{\rho}[f(\mathbf{x},\cdot)]]$. *Therefore, to compute the RTS performance of a set of inputs, one just has to compute the corresponding R2 utility of the union risk-adjusted output set. In general, a similar result does not hold for the STR-R2 utility unless the operations commute (Section 3.3.3).*

5.2 Robust inverted generational distance

The inverted generational distance (IGD) is a popular multi-objective performance metric that was initially introduced by Coello and Reyes Sierra [2004]. Following Schutze et al. [2012],

23

the IGD indicator $I^{\mathrm{IGD}_{p,q}} : 2^{\mathbb{R}^M} \times B[\mathbb{R}^M] \to \mathbb{R}$ computes some notion of the average distance between a set of vectors $Y \subset \mathbb{R}^M$ and a finite set of ideal (or utopia) vectors $\Upsilon \in \mathbb{B}[\mathbb{R}^M]$,

$$I^{\mathrm{IGD}_{p,q}}[Y, \Upsilon] = \left(\frac{1}{|\Upsilon|} \sum_{\boldsymbol{v} \in \Upsilon} \left(\inf_{\mathbf{y} \in Y} ||\boldsymbol{v} - \mathbf{y}||_{L^p} \right)^q \right)^{1/q}, \tag{27}$$

for some norms $p, q \geq 1$, where $\mathbb{B}[\mathbb{R}^M] := \{Y \subseteq \mathbb{R}^M : |Y| < \infty\}$ denotes the space of finite sets of vectors. By construction, a smaller IGD indicator value is more desirable. Following Tu et al. [2025], we can rewrite the IGD indicator as a transformation of an R2 utility, namely the IGD utility

$$U^{\mathrm{IGD}_{p,q}}[Y] := -(I^{\mathrm{IGD}_{p,q}}[Y, \Upsilon])^q = \frac{1}{|\Upsilon|} \sum_{\boldsymbol{v} \in \Upsilon} \sup_{\mathbf{y} \in Y} s_{\boldsymbol{v}}^{\mathrm{IGD}_{p,q}}(\mathbf{y}), \tag{28}$$

where $s_{\boldsymbol{v}}^{\mathrm{IGD}_{p,q}}(\mathbf{y}) = -||\boldsymbol{v} - \mathbf{y}||_{L^p}^q$ is the IGD scalarisation function. Note that the total ordering imposed by the IGD utility (28) remains the same as the IGD indicator (27) but with the directions reversed. That is, a smaller indicator value was preferred before, whereas now, a larger utility value is more desirable. By substituting this R2 utility set-up into the corresponding expressions for the robust R2 utilities (25) and (26), we obtain the following robust generalisations of the IGD utility:

$$R_{f,\boldsymbol{\rho}}^{\mathrm{RTS\text{-}IGD}_{p,q}}[X] := \frac{1}{|\Upsilon|} \sum_{\boldsymbol{v} \in \Upsilon} \sup_{\mathbf{x} \in X} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]} s_{\boldsymbol{v}}^{\mathrm{IGD}_{p,q}}(\mathbf{y}),$$

$$R_{f,\rho}^{\mathrm{STR\text{-}IGD}_{p,q}}[X] := \frac{1}{|\Upsilon|} \sum_{\boldsymbol{v} \in \Upsilon} \sup_{\mathbf{x} \in X} \rho[s_{\boldsymbol{v}}^{\mathrm{IGD}_{p,q}}(f(\mathbf{x}, \cdot))],$$

which holds for any set of inputs $X \subseteq \mathbb{X}$ and norms $p, q \geq 1$. In other words, the RTS-IGD utility computes the average negative distance between the risk-adjusted output set and the set of utopia points, whilst the STR-IGD utility computes the average risk-adjusted negative distance between the distribution of possible objective vectors and the set of utopia points. On the left of Figure 6, we give a pictorial illustration of these different computations on a simple two-dimensional example.

Remark 5.2 (Robust IGD+ utility) *We can also generalise the above discussion for the corresponding IGD+ indicator described by Ishibuchi et al. [2015]. This is a straightforward extension, where the IGD scalarisation function is replaced with its truncated version [Tu et al., 2025].*
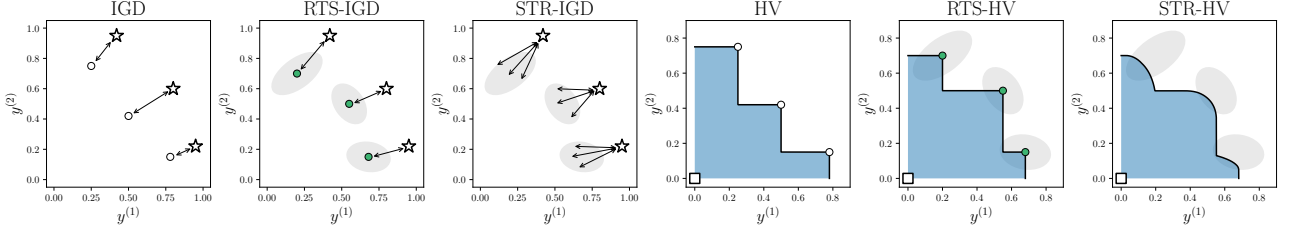
5.3 Robust hypervolume indicator

The hypervolume indicator (HV) is another popular multi-objective performance metric which was proposed by Zitzler and Thiele [1998]. Formally, the hypervolume indicator $I^{\mathrm{HV}} : 2^{\mathbb{R}^M} \times \mathbb{R}^M \to \mathbb{R}$ computes the volume that is enclosed between a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$ and the Pareto front surface of a set of vectors $Y \subset \mathbb{R}^M$, that is

$$I^{\mathrm{HV}}[Y, \boldsymbol{\eta}] := \int_{\mathbb{R}^M} \mathbb{1}[\mathbf{z} \in \mathbb{D}_{\preceq}[\mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[Y]] \cap \mathbb{D}_{\succcurlyeq}[\{\boldsymbol{\eta}\}]]d\mathbf{z}. \tag{29}$$

As identified in existing work [Shang et al., 2018, Deng and Zhang, 2019, Zhang and Golovin, 2020], we can rewrite the hypervolume indicator as an R2 utility,

$$U_{\boldsymbol{\eta}}^{\mathrm{HV}}[Y] := I_{\mathrm{HV}}[Y, \boldsymbol{\eta}] = \mathbb{E}_{\boldsymbol{\lambda} \sim \mathrm{Uniform}(\mathcal{S}_+^{M-1})} \left[\sup_{\mathbf{y} \in Y} \tau^{\mathrm{HV}}(s_{(\boldsymbol{\eta}, \boldsymbol{\lambda})}^{\mathrm{Len}}(\mathbf{y})) \right],$$

Figure 6: *An illustration of the standard and robust IGD (Section 5.2) and hypervolume (Section 5.3) indicators on a two-dimensional example. In the plots, we illustrate the output associated with three inputs which are assumed to be either fixed for the standard (no uncertainty) examples, or uniformly distributed on some ellipsoid for the robust examples. For the robust utilities, we used the expectation functionals; for the IGD indicator, we use three utopia points (stars); and for the HV indicator, we set the reference point to the origin (square).*

where $\tau^{\mathrm{HV}}(x) = c_M x^M$ is the hypervolume transformation function which depends on a constant $c_M = \pi^{M/2}/(2^M \Gamma(M/2+1))$ with $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ denoting the Gamma function. As we showcase below in Proposition 5.1 and Proposition 5.2, the hypervolume enclosed between a reference vector and the RTS and STR fronts can also naturally be written as an RTS-R2 and STR-R2 utility, respectively. We prove both of these results in Appendix A.11. For some additional intuition, we also present a two-dimensional example of these different hypervolumes on the right of Figure 6.

Proposition 5.1 (RTS hypervolume) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, a bounded multivariate risk functional $\boldsymbol{\rho}$ and a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. The hypervolume of the corresponding RTS front (22) can be written as an RTS-R2 utility*

$$R_{\boldsymbol{\eta},f,\boldsymbol{\rho}}^{\mathrm{RTS\text{-}HV}}[X] := \int_{\mathbb{R}^M} \mathbb{1}[\mathbf{z} \in \mathbb{D}_{\preceq}[\mathcal{F}_{\boldsymbol{\eta},f,\boldsymbol{\rho}}^{\mathrm{RTS}}[X]] \cap \mathbb{D}_{\succ\succ}[\{\boldsymbol{\eta}\}]]d\mathbf{z}$$

$$= \mathbb{E}_{\boldsymbol{\lambda} \sim \mathrm{Uniform}(\mathcal{S}_+^{M-1})} \left[\sup_{\mathbf{x} \in X} \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x},\cdot)]} \tau^{\mathrm{HV}}(s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(\mathbf{y})) \right].$$

Proposition 5.2 (STR hypervolume) *Consider a bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, a partially coherent univariate risk functional ρ and a reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. The hypervolume of the corresponding STR front (23) can be written as an STR-R2 utility*

$$R_{\boldsymbol{\eta},f,\rho}^{\mathrm{STR\text{-}HV}}[X] := \int_{\mathbb{R}^M} \mathbb{1}[\mathbf{z} \in \mathbb{D}_{\preceq}[\mathcal{F}_{\boldsymbol{\eta},f,\rho}^{\mathrm{STR}}[X]] \cap \mathbb{D}_{\succ\succ}[\{\boldsymbol{\eta}\}]]d\mathbf{z}$$

$$= \mathbb{E}_{\boldsymbol{\lambda} \sim \mathrm{Uniform}(\mathcal{S}_+^{M-1})} \left[\sup_{\mathbf{x} \in X} \rho^{\mathrm{HV}}[s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(f(\mathbf{x},\cdot))] \right]$$

where $\rho^{\mathrm{HV}} := \tau^{\mathrm{HV}} \circ \rho$.

Remark 5.3 (Monte Carlo estimate) *Computationally, the hypervolume indicators defined above can be estimated efficiently using a Monte Carlo estimate with $J > 0$ samples of the positive unit vectors $\boldsymbol{\lambda}_j \sim \mathrm{Uniform}(\mathcal{S}_+^{M-1})$. For instance, one can estimate the STR hypervolume with the equation*

$$\hat{R}_{\boldsymbol{\eta},f,\rho,J}^{\mathrm{STR\text{-}HV}}[X] = \frac{1}{J} \sum_{j=1}^J \sup_{\mathbf{x} \in X} \rho^{\mathrm{HV}}[s_{(\boldsymbol{\eta},\boldsymbol{\lambda}_j)}^{\mathrm{Len}}(f(\mathbf{x},\cdot))],$$

for any set of inputs $X \subseteq \mathbb{X}$, bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, partially coherent univariate risk functional ρ and reference vector $\boldsymbol{\eta} \in \mathbb{R}^M$. By a simple calculation, we see that the variance of this estimate can be uniformly upper bounded by a function dependent constant

25

divided by the number of Monte Carlo samples, that is

$$\mathbb{V}\mathrm{ar}[\hat{R}^{\mathrm{STR\text{-}HV}}_{\boldsymbol{\eta},f,\rho,J}[X]] \leq \frac{1}{J}\mathbb{E}_{\boldsymbol{\lambda}\sim\mathrm{Uniform}(\mathcal{S}^{M-1}_+)}\left[\sup_{\mathbf{x}\in X}\rho^{\mathrm{HV}}[s^{\mathrm{Len}}_{(\boldsymbol{\eta},\boldsymbol{\lambda})}(f(\mathbf{x},\cdot))]^2\right]$$

$$\leq \frac{c^2_M}{J}\sup_{(\mathbf{x},\boldsymbol{\xi})\in\mathbb{X}\times\Xi}||f(\mathbf{x},\boldsymbol{\xi})-\boldsymbol{\eta}||^{2M}_{L^2}\cdot\mathbb{1}[f(\mathbf{x},\boldsymbol{\xi})\in\mathbb{D}_{\nrightarrow}[\{\boldsymbol{\eta}\}]].$$

Similarly, one can also devise and bound the variance of the Monte Carlo estimate for the standard and RTS hypervolume indicators in the same way.

6 Numerical examples

We now present two numerical case studies in order to highlight the efficacy of the novel ideas that we have presented in this work. Firstly, in Section 6.1, we present the cake baking case study which illustrates a practical example where one has to decide between an RTS and STR solution. Secondly, in Section 6.2, we present the robust rocket injector case study which showcases how one can use the concepts of a robust Pareto front and a robust performance metric in order to devise and assess the performance of some robust multi-objective optimisation algorithms.

6.1 Cake problem

In this numerical example, we illustrate the utility of the RTS and STR approach through a real-world case study. This analysis is based on the cake data set $\mathcal{D}_T$, which is a public[4] data set describing the results of $T = 50$ cake baking experiments. Formally, this data set investigates how $M = 3$ properties of a cake, described by the multi-objective function $g : \mathbb{X} \to \mathbb{R}^M$,

$$g(\mathbf{x}) = (-\mathrm{CALORIES}(\mathbf{x}), \mathrm{TASTE}(\mathbf{x}), \mathrm{BROWNING}(\mathbf{x})) \in \mathbb{R}^M,$$

changes when the relative composition of its $D = 6$ main ingredients are varied. The inputs for this problem are the relative amount of ingredients $\mathbf{x} \in \mathbb{X}$, which lie in the $(D-1)$-dimensional simplex and satisfy the following inequality constraints:

$$\mathbb{X} = C_1 \cap C_2 \cap C_3 \cap C_4,$$
$$C_1 = \{\mathbf{x} \in \mathbb{R}^D_{\geq 0} : ||\mathbf{x}||_{L^1} = 1\},$$
$$C_2 = \{\mathbf{x} \in \mathbb{R}^D : x^{(1)} + x^{(2)} \in [0.2, 0.4]\},$$
$$C_3 = \{\mathbf{x} \in \mathbb{R}^D : x^{(3)} \in [0.15, 0.35]\},$$
$$C_4 = \{\mathbf{x} \in \mathbb{R}^D : x^{(4)} + x^{(5)} + x^{(6)} \geq 0.2\}.$$

Intuitively, these constraints are enforced in order to ensure that the resulting bakes lead to valid cakes. In words, condition one (C_1) is the simplex constraint; condition two (C_2) is a constraint on the amount of flour that is used within the bake; condition three (C_3) is a constraint on the amount of sugar that is used; and lastly condition four (C_4) is a constraint on the amount of chocolate, nuts and carrots that is used within the mix. In Figure 7, we illustrate this data set and highlight the corresponding $P = 14$ observed Pareto optimal points.

In practice, given a data set such as this, one wishes to solve the inverse problem. That is, to find a cake whose properties achieves a certain set of desirable values. As this data set only contains a finite number of experiments, we follow standard procedure and fit a probabilistic

---

[4]The cake data set is available in the following Github repository: https://github.com/basf/mopti.
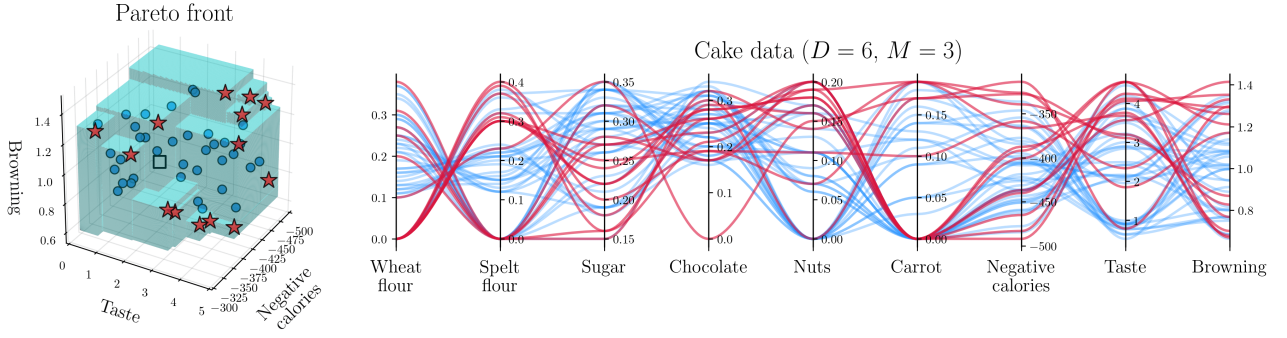
Figure 7: *An illustration of the cake data set. On the left, we plot the Pareto front surface of the $T = 50$ observations. On the right, we draw the corresponding parallel coordinates plot over both the input and output space. The sub-optimal points are coloured in blue, whilst the Pareto optimal points are highlighted in red.*

model on it. Intuitively, this model is used as a surrogate in order to get a prediction on the properties of a cake at any feasible set of ingredients. For convenience, in this example, we fit a Gaussian process model, which is a popular model that is often used in industrial applications—see for instance Rasmussen and Williams [2006] for a review. More precisely, we let $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$, denote the corresponding posterior Gaussian process model that was fitted on the data points. That is, every instance of the uncertain parameter $\boldsymbol{\xi} \in \Xi$ corresponds to a different sample path of the Gaussian process model:

$$(f(\mathbf{x}_1, \boldsymbol{\xi}), \dots, f(\mathbf{x}_N, \boldsymbol{\xi})) \sim \mathcal{N}(\boldsymbol{\mu}_T(X), \boldsymbol{\Sigma}_T(X, X))$$
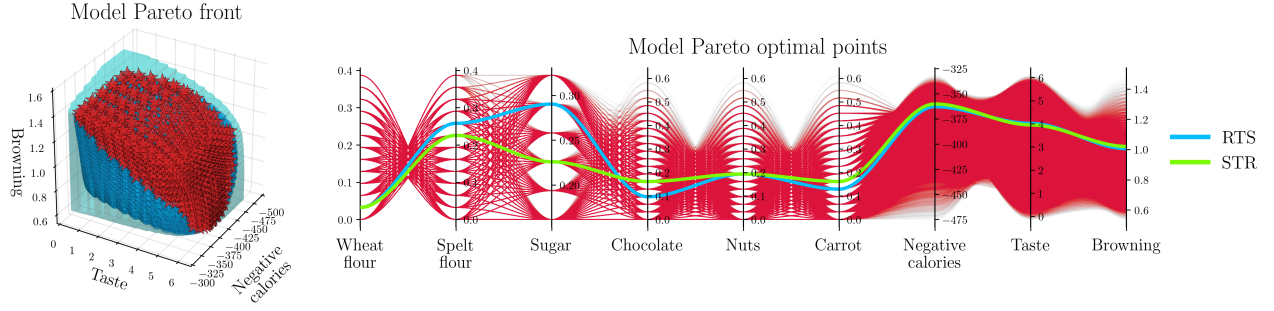
for any finite set of inputs $X = (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{X}^N$, where $\boldsymbol{\mu}_T : \mathbb{X} \to \mathbb{R}^M$ and $\boldsymbol{\Sigma}_T : \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{M \times M}$ denotes the corresponding mean and covariance function of the posterior Gaussian process model. In Figure 8, we plot the posterior mean of this model over a discretisation of the input space with $|\hat{\mathbb{X}}| = 44835$. We also highlight the corresponding Pareto optimal vectors and Pareto front surface.

Suppose for instance that the decision maker is interested in identifying the ingredients that are required to make a cake whose properties achieve a target value of $\boldsymbol{v} = (-360, 4, 1) \in \mathbb{R}^M$. As motivated earlier in Example 3.1, one can recast this target optimisation problem as a robust scalarised optimisation problem. Specifically, one can use the expectation functionals (Examples 2.3 and 2.8) along with a distance-based scalarisation function. As a concrete example, we consider using the following weighted variant of the IGD scalarisation function:

$$s_{(\boldsymbol{v}, \mathbf{w})}^{\text{WIGD}_{2,2}}(\mathbf{y}) := -||\mathbf{w}(\boldsymbol{v} - \mathbf{y})||_{L^2}^2 := -\sum_{m=1}^{M} \left(w^{(m)}(v^{(m)} - y^{(m)}) \right)^2$$

for some weight vector $\mathbf{w} \in \Delta^{M-1}$. The introduction of this weight vector is mainly to ensure that each objective is normalised so that they have similar ranges. In particular, it ensures that the distance computation is not skewed towards the objectives with larger ranges such as the negative number of calories. For our example, we set the weight to be the normalised inverse of the estimated objective ranges: $\mathbf{w} = (0.004, 0.155, 0.841) \in \Delta^{M-1}$. Equipped with this set-up, the corresponding RTS and STR solutions for this target optimisation problem reduces down to the following two expressions:

$$\mathbf{x}_{\text{RTS}}^* \in \underset{\mathbf{x} \in \mathbb{X}}{\arg\min} \, ||\mathbf{w}(\boldsymbol{v} - \boldsymbol{\mu}_T(\mathbf{x}))||_{L^2}^2,$$

$$\mathbf{x}_{\text{STR}}^* \in \underset{\mathbf{x} \in \mathbb{X}}{\arg\min} \left(||\mathbf{w}(\boldsymbol{v} - \boldsymbol{\mu}_T(\mathbf{x}))||_{L^2}^2 + ||\mathbf{w}\sqrt{\text{DIAGONAL}(\boldsymbol{\Sigma}_T(\mathbf{x}, \mathbf{x}))}||_{L^2}^2 \right),$$

27

Figure 8: *An illustration of the mean function $\boldsymbol{\mu}_T : \mathbb{X} \to \mathbb{R}^M$ associated with a Gaussian process model that was trained on the cake data set. On the left, we plot the mean vectors over a discretisation of the input space and its corresponding Pareto front surface. On the right, we draw a parallel coordinates plot of the Pareto optimal vectors in red. In addition, we also highlight the RTS and STR solution for the target optimisation problem (Section 6.1) in light blue and green, respectively.*

where all operations on vectors are defined component-wise and the DIAGONAL operation is used in order to extract the M-dimensional vector of variances from the square covariance matrix $\boldsymbol{\Sigma}_T(\mathbf{x}, \mathbf{x}) \in \mathbb{R}^{M \times M}$ for any input $\mathbf{x} \in \mathbb{X}$. Evidently, from these equations, we see that the difference between an RTS and STR approach is very similar to the classical bias-variance trade-off. The RTS problem is solely focussed on minimising the bias. That is, it tries to seek out the set of ingredients whose expected output is as close to the target as possible in terms of the weighted L^2-norm. In contrast, the STR approach also considers the effect of the variance in the scalarised problem as well. In particular, the STR problem contains an additional variance penalty term which penalises points whose weighted variances are large. In Figure 8, we plot the corresponding RTS and STR solution for the discretised problem:

$$\mathbf{x}^*_{\mathrm{RTS}} = (0.0323, 0.2581, 0.2903, 0.0968, 0.1935, 0.1290),$$
$$\mathbf{x}^*_{\mathrm{STR}} = (0.0323, 0.2258, 0.2258, 0.1613, 0.1935, 0.1613).$$

Notably, the main difference between these two recipes lies in the relative amount of sugar that is used. The RTS approach suggests that we should have a recipe that uses more sugar if we are interested in getting an expected performance that is close to the target. In contrast, the STR approach suggests that we should substitute the additional sugar in favour of more chocolate and carrots. Conceptually, we should use the RTS solution over the STR solution in the aggregation scenario, where we assess the performance based on a collective aggregate. For instance, if we were making and selling boxes of little cupcakes, then this RTS solution would be more ideal because we would be interested in optimising the average quality in each box. In contrast, the STR solution becomes more useful in the scenario where the quality of each cake is assessed individually. For instance, if we were selling many individual cakes, then we should favour the STR solution as the quality of each cake should be assessed independently in this setting.

6.2 Robust rocket injector problem

In this numerical example, we study a distributionally robust variant of the rocket injector problem. This problem was initially formulated in the work by Vaidyanathan et al. [2003] and was later implemented by Tanabe and Ishibuchi [2020]. Formally, this problem studies the optimisation of $M = 3$ (scaled) objectives relating to the performance and longevity of a rocket injector design $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$,

$$f(\mathbf{x}, \boldsymbol{\xi}) = (-\text{MaxFaceTemp}(\mathbf{x}, \boldsymbol{\xi}), -\text{CombustionLength}(\mathbf{x}, \boldsymbol{\xi}), -\text{MaxOxTipTemp}(\mathbf{x}, \boldsymbol{\xi})) \in \mathbb{R}^M,$$

28

subject to changes in $D+W = 4$ geometric features $(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{X} \times \Xi$. Unlike the original work, we consider a robust variant of this problem in which only a subset of the geometric features can be adjusted by the decision maker. The other components are assumed to be determined by the supply manufacturer who provides a base construction which is subject to some uncertainty. In particular, we suppose that we can control $D = 2$ geometric features lying in the normalised space $\mathbb{X} = [0,1]^D$, whilst the manufacturer can control the other $W = 2$ pertinent features lying in the normalised space $\Xi = [0,1]^W$. Explicitly, the controllable features are the ones affecting the Hydrogen supply, whilst the uncontrollable features affect the Oxygen supply:

$x^{(1)}$: Normalised angle at which the H_2 is directed toward the oxidiser,

$x^{(2)}$: Normalised change in the H_2 flow area from the baseline,

$\xi^{(1)}$: Normalised change in the O_2 flow area from the baseline,

$\xi^{(2)}$: Normalised oxidiser post-tip thickness.

For the manufacturer's distribution, we assume a uniform distribution $p(\boldsymbol{\xi}) \sim \text{Uniform}(\Xi)$ over the uncertain parameters $\boldsymbol{\xi} \in \Xi$. For our robust objective, we consider an STR approach where the goal is to identify the STR robust designs whose output distribution lies on the CVaR Pareto front surface at level $\alpha = 0.9$. Equivalently, we formulate this problem as a robust utility optimisation problem where we are interested in optimising the STR hypervolume (Proposition 5.2) of the identified CVaR front

$$\max_{X \subseteq \mathbb{X}, |X| \leq T} R^{\text{STR-HV}}_{\boldsymbol{\eta}, f, \rho^{\text{CVaR}_\alpha}}[X],$$

where the reference vector $\boldsymbol{\eta} = (-1.05, -1.30, -1.20) \in \mathbb{R}^M$ is set to a lower bound of the objective function and $T > 0$ is a cardinality constraint. Note that the inclusion of this cardinality constraint is crucial because it encodes the fact that we are interested in finding a finite set of high quality points as opposed to a large set of potentially variable quality points. In fact, without this constraint, one can trivially solve this problem by proposing the whole input space $X = \mathbb{X}$ because the hypervolume is monotonic and therefore there is no loss in utility with adding more points.

To emulate a real-world scenario, we treat the objective function f as a black-box function that can only be evaluated sparingly at a few carefully selected design locations $T = 65$. More precisely, at any time $t > 0$, the decision maker is given the opportunity to see a noisy function evaluation at any chosen design $\mathbf{x}_t \in \mathbb{X}$, that is $\mathbf{y}_t = f(\mathbf{x}_t, \boldsymbol{\xi}_t) + \boldsymbol{\epsilon}_t \in \mathbb{R}^M$, where $\boldsymbol{\xi}_t \sim p(\boldsymbol{\xi})$ is the uncertain parameter that was independently sampled by the manufacturer and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}_M, \text{DIAG}(\boldsymbol{\sigma}^2))$ is the observation noise. In our example, we consider a zero-mean Gaussian observation noise whose standard deviation $\boldsymbol{\sigma} = (0.01, 0.01, 0.01)$ is set to 1% of the estimated scaled objective range. To initialise this problem, we randomly sampled $T_0 = 5$ initial points from the controllable input space. In Figure 9, we illustrate these initial points in the objective space and plot the corresponding CVaR front under the STR approach. Notably, to compute this CVaR front, we require knowledge of the true objective function at each uncertain parameter, which is typically not available in a real-world set-up.

To approximately solve this robust multi-objective optimisation problem, we require an acquisition procedure in order to sequentially select the inputs to evaluate at. Note that we cannot deploy the STR approach directly for this problem because we do not have access to the true objective function. Instead, we propose using a straightforward extension of the standard hypervolume improvement acquisition function [Emmerich et al., 2006]. More specifically, at each time $t > 0$ we propose choosing the input $\mathbf{x}_t \in \mathbb{X}$ which maximises the STR hypervolume
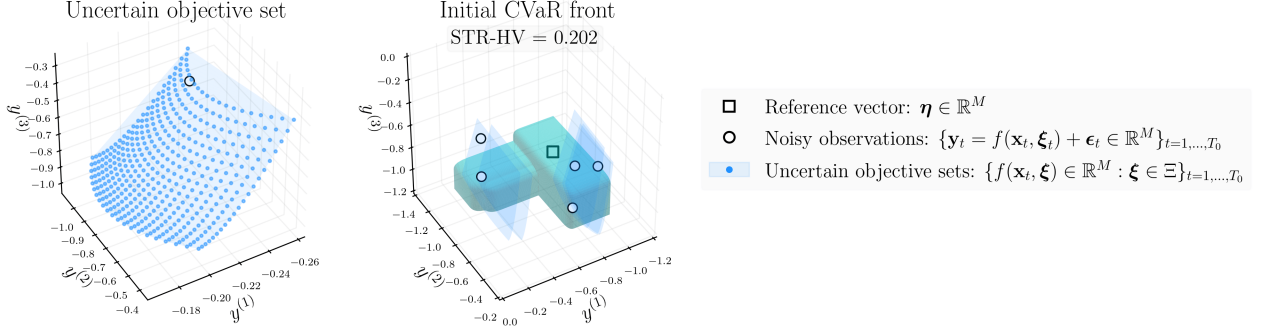
Figure 9: *This plot illustrates the initial CVaR front at level $\alpha = 0.9$ for the $T_0 = 5$ initial designs under the STR approach. On the left, we plot the uncertainty objective set associated with a single input location. On the right, we plot the CVaR front associated with the initial set of inputs.*
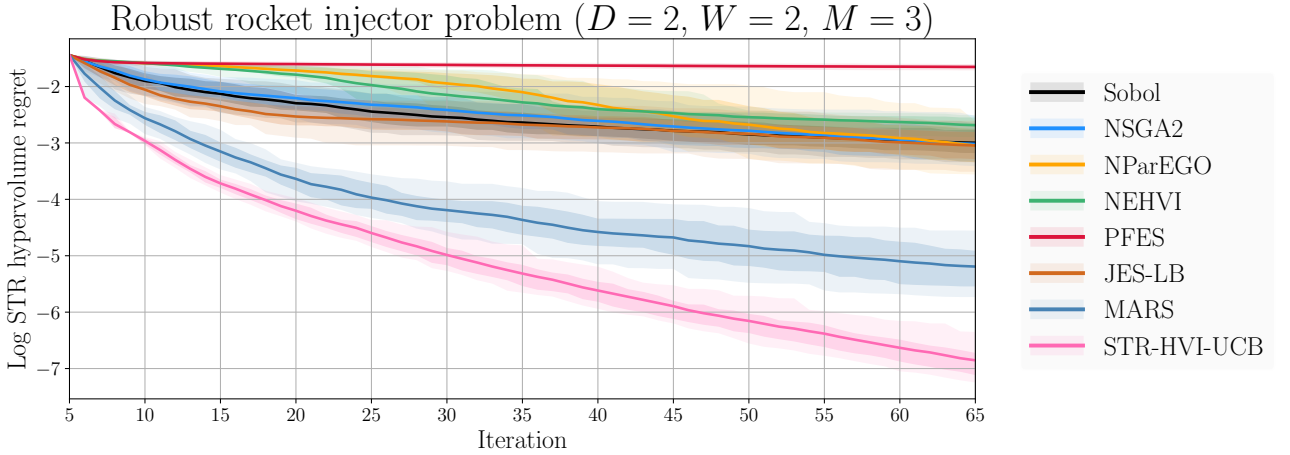
Figure 10: *A numerical comparison between the different multi-objective algorithms on the robust rocket injector problem over 100 independent runs. For each method, we plot the mean log regret at every iteration along with an uncertainty estimate. Specifically, we highlight the band between the 0.25 to 0.75 and 0.1 to 0.9 quantiles with a dark and light shade, respectively.*

improvement under the latest surrogate model

$$\mathbf{x}_t \in \arg\max_{\mathbf{x} \in \mathbb{X}} R^{\text{STR-HV}}_{\hat{\boldsymbol{\eta}}_{t-1}, \hat{f}_{t-1}, \rho^{\text{CVaR}_\alpha}}[X_{t-1} \cup \{\mathbf{x}\}] - R^{\text{STR-HV}}_{\hat{\boldsymbol{\eta}}_t, \hat{f}_{t-1}, \rho^{\text{CVaR}_\alpha}}[X_{t-1}],$$

where $X_{t-1} = \{\mathbf{x}_n\}_{n \leq t-1}$ denotes the set of previously queried inputs, whilst $\hat{\boldsymbol{\eta}}_t \in \mathbb{R}^M$ and $\hat{f}_{t-1} : \mathbb{X} \times \Xi \to \mathbb{R}^M$ denotes the latest estimate of the reference vector and objective function, respectively. In our example, we set the latest reference vector to be an over-estimation of the lower bound $\eta^{(m)}_{t-1} = \min_{n<t} y^{(m)}_n - \kappa(\max_{n<t} y^{(m)}_n - \min_{n<t} y^{(m)}_n)$ with $\kappa = 0.1$ for $m = 1, \ldots, M$. For the surrogate model, we used an upper confidence bound of a Gaussian process model which was trained on the previous observations, that is $\hat{f}^{\text{UCB}}_{t-1}(\mathbf{x}, \boldsymbol{\xi}) = \boldsymbol{\mu}_{t-1}(\mathbf{x}, \boldsymbol{\xi}) + \beta \boldsymbol{\sigma}_{t-1}(\mathbf{x}, \boldsymbol{\xi})$, where $\beta = 2.0$ is the trade-off parameter and the vectors $\boldsymbol{\mu}_{t-1}(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{R}^M$ and $\boldsymbol{\sigma}_{t-1}(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{R}^M_{\geq 0}$ are the mean and standard deviation of the model at the input pair $(\mathbf{x}, \boldsymbol{\xi}) \in \mathbb{X} \times \Xi$. We refer to this overall acquisition strategy as the STR-HVI-UCB approach.

In Figure 10, we present the results of our numerical comparison in which we compared our proposed method versus the existing baselines over 100 independent runs on a discretised version of the robust rocket injector problem. Namely, we compare our method against non-robust[5]

---

[5]These approaches are non-robust in the sense that they can condition on the observed data set $\mathcal{D}_{t-1} =$
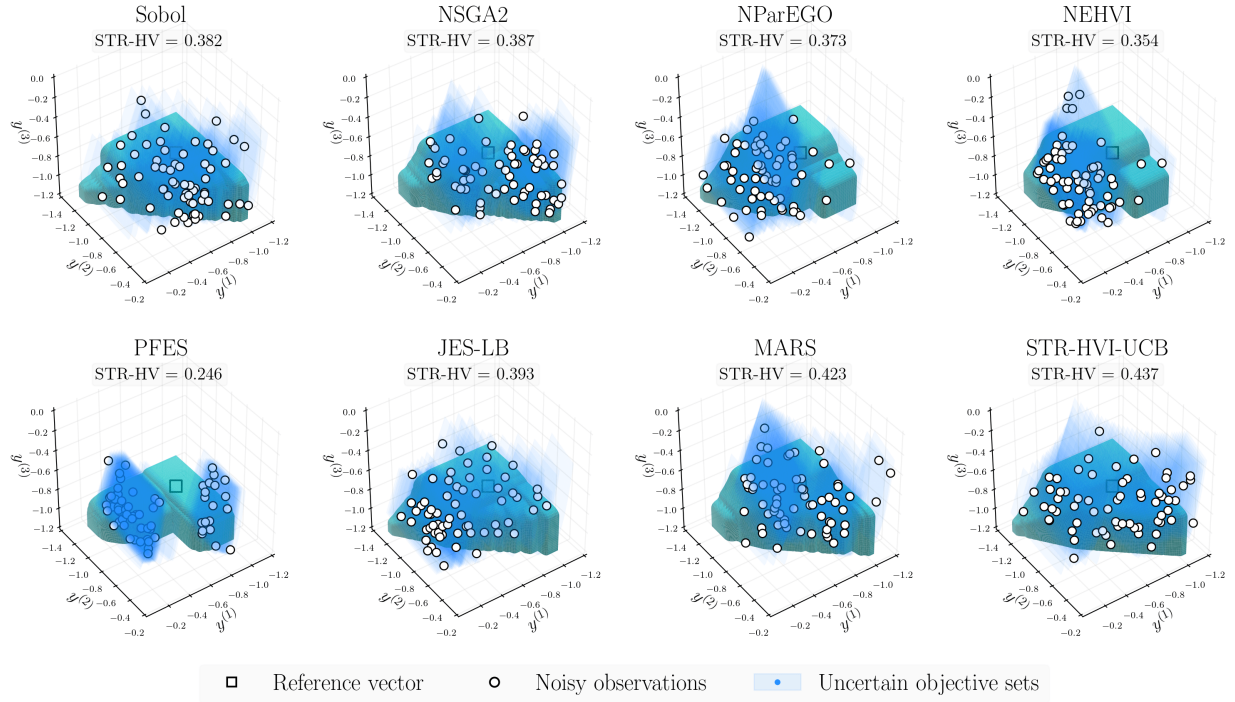
Figure 11: *An illustration of the final CVaR fronts identified by the different multi-objective optimisation algorithms on the robust rocket injector problem for one experimental run with $T = 65$ function evaluations. The maximum hypervolume on this example is around $R^{\text{STR-HV}}_{\boldsymbol{\eta}, f, \rho^{\text{CVaR}_\alpha}}[\mathbb{X}] \approx 0.4382$.*

approaches such as: the random search (Sobol), NSGA2 [Deb et al., 2002], NParEGO [Knowles, 2006, Daulton et al., 2021], NEHVI [Emmerich et al., 2006, Daulton et al., 2021], PFES [Suzuki et al., 2020] and the JES-LB [Tu et al., 2022] algorithms. We also compare our approach against the MARS algorithm [Daulton et al., 2022], which is a robust algorithm that takes an STR approach using the Chebyshev scalarisation function and the univariate VaR risk functional. Unsurprisingly we see that the robust methods perform better than the non-robust methods. Moreover, our method performs the best overall under the chosen utility. For some visual intuition, we have included Figure 11 where we plot the final CVaR front obtained by each algorithm under the same random seed. Visually we see that most of the variation between these algorithms transpire around the first and second objectives. The existing approaches regularly struggle to find the inputs which robustly maximises the CVaR front over these two objectives. In contrast, our robust approach is much more consistent and is often very quick in identifying the difficult inputs which robustly maximise the whole CVaR front.

7   Conclusion

In this work, we studied the robust multi-objective optimisation problem from a computational perspective. We identified that the majority of existing robust multi-objective optimisation algorithms, in one way or another, rely on two key operations: robustification and scalarisation. These two procedures are in general not commutative and therefore the order that they are applied in has an effect on the resulting solutions that are identified and the final decisions that are made. As part of our work, we analysed the effect of this ordering from both a philosophical and a methodological perspective. Notably, among other things, we proposed the novel concepts of a robust Pareto front surface and a robust performance metric. We then demonstrated the

---

$\{(\mathbf{x}_n, \boldsymbol{\xi}_n, \mathbf{y}_n)\}_{n \leq t-1}$, at every time t, but they do not consider the uncertainty present in the uncontrollable variable $\boldsymbol{\xi} \in \Xi$.

practical utility of these new ideas via some insightful numerical experiments.

In closing, the key ideas and methodology which we presented in this work are very general and encapsulates many existing ideas and algorithms into a neat computational framework. Future work could focus on refining these ideas further for specific problems and case studies. For instance, practitioners in industry are often interested in finding and using optimisation algorithms which are both numerically performant and theoretically justified. Many such algorithms and results has been developed for the robust single-objective optimisation problem over the past decades [Ben-Tal et al., 2009, Bertsimas et al., 2011, Rahimian and Mehrotra, 2019]. Extending these computational algorithms and theoretical results to robust multi-objective setting is an interesting and practically useful direction for future research.

Acknowledgements

A Proofs

A.1 Proof of Theorem 2.1

The proof of this result is presented by Miettinen [1998, Part 2, Theorem 3.4.5]

∎

A.2 Proof of Lemma 2.1

Consider any partially coherent univariate risk functional ρ and bounded function $h : \Xi \to \mathbb{R}$. We will show that $\rho[h]$ is bounded and therefore ρ is a bounded univariate risk functional. As h is bounded, there exists constants $l, u \in \mathbb{R}$ such that $l \leq h(\boldsymbol{\xi}) \leq u$ for all $\boldsymbol{\xi} \in \Xi$. Let $r \in \mathbb{R}$ be any constant and define $c_r : \Xi \to \mathbb{R}$ to be the corresponding constant function satisfying $c_r(\boldsymbol{\xi}) = r$ for all $\boldsymbol{\xi} \in \Xi$. Then by the normalised and translation equivariance property, we have that $\rho[c_r] = r$. By the monotonicity property, we have that $\rho[c_l] \leq \rho[h] \leq \rho[c_u]$. Together, these two results imply that $\rho[h]$ is bounded since $l \leq \rho[h] \leq u$.

∎

A.3 Proof of Proposition 3.1

Consider any weakly union robust input $\mathbf{z} \in \mathbb{X}$ for the union robust optimisation problem (10). By the definition of weak union robustness, there must exist an output vector $\mathbf{y}^* \in \boldsymbol{\rho}[f(\mathbf{z}, \cdot)]$ which is weakly optimal. We will now proceed to show that the input $\mathbf{z}$ must be a maximiser to the RTS length scalarised optimisation problem (9) with the positive direction $\boldsymbol{\lambda} = (\mathbf{y}^* - \boldsymbol{\eta})/||\mathbf{y}^* - \boldsymbol{\eta}||_{L^2} \in \mathcal{S}_+^{M-1}$. Suppose for a contradiction that there exists a distinctive input $\mathbf{x} \in \mathbb{X}$ which achieves a greater RTS length scalarised value than the input $\mathbf{z}$ along this positive direction. This implies that

$$
\max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]} \min_{m=1,\ldots,M} \frac{y^{(m)} - \eta^{(m)}}{\lambda^{(m)}} > \max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{z}, \cdot)]} \min_{m=1,\ldots,M} \frac{y^{(m)} - \eta^{(m)}}{\lambda^{(m)}} = ||\mathbf{y}^* - \boldsymbol{\eta}||_{L^2}
$$

due to the weak optimality of $\mathbf{y}^*$ and the strong monotonicity of the length scalarisation function [Tu et al., 2024, Lemma A.2]. This further implies that

$$
\max_{\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]} \min_{m=1,\ldots,M} \frac{y^{(m)} - \eta^{(m)}}{y^{*(m)} - \eta^{(m)}} > 1,
$$

which means that there exists a vector $\mathbf{y} \in \boldsymbol{\rho}[f(\mathbf{x}, \cdot)]$ that strongly dominates $\mathbf{y}^*$. This contradicts the weak optimality of $\mathbf{y}^*$ and therefore concludes the proof.

∎

A.4 Proof of Theorem 4.1

The proof of this result is presented by Tu et al. [2024, Appendix A.1].

∎

A.5 Pareto front conditions

In order to verify whether a polar surface is a Pareto front surface, we will rely on the key result by Tu et al. [2024, Proposition 3.1], which we recall below in Proposition A.1.

Proposition A.1 (Pareto front conditions) *[Tu et al., 2024, Proposition 3.1] Consider a polar surface $A \in \mathbb{L}_{\boldsymbol{\eta}}$, then this set is a Pareto front surface, that is $A = \mathcal{Y}_{\boldsymbol{\eta}}^{\mathrm{int}}[A]$, if and only if the following two conditions holds:*

C1. The positive lengths condition: $\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[A] > 0$ for all $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$.

C2. The maximum ratio condition: $\max_{m=1,\ldots,M} \frac{\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[A]\lambda^{(m)}}{\ell_{\boldsymbol{\eta},\boldsymbol{v}}[A]v^{(m)}} \geq 1$ for all $\boldsymbol{\lambda}, \boldsymbol{v} \in \mathcal{S}_+^{M-1}$.

A.6   Proof of Proposition 4.1

Consider the Pareto front surface statistic defined in (21). Firstly, we will show that this set is indeed a polar surface. By definition, to establish that this set is a polar surface, we need to establish that the projected lengths

$$\rho[\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\cdot)]] = \rho\left[\sup_{\mathbf{x}\in\mathbb{X}} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(f(\mathbf{x},\cdot))\right] \tag{30}$$

are non-negative and bounded along every positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. This result follows directly from the definition of the length scalarisation function (5) and the boundedness of the partially coherent univariate risk functional (Lemma 2.1). We will now show that this polar surface is either a Pareto front surface or the degenerate singleton set. The former setting (Case 1) happens whenever there exists no positive direction where the projected length is zero. The latter setting (Case 2) happens whenever there exists at least one positive direction where the projected length is zero. Below, we consider these two cases separately.

Case 1:   Suppose that the projected length (30) is positive along all positive directions. Then the projected length condition (C1) holds. We will now show that the maximum ratio condition (C2) also holds and therefore the polar surface (21) is a Pareto front surface. Note that for any $\boldsymbol{\xi} \in \Xi$, the set $Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi})$ (20) is either a valid Pareto front surface or a degenerate singleton set. In either case, it satisfies the maximum ratio condition

$$\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi})] \max_{m=1,\ldots,M} \frac{\lambda^{(m)}}{v^{(m)}} \geq \ell_{\boldsymbol{\eta},\boldsymbol{v}}[Y_{\boldsymbol{\eta},f}^*(\boldsymbol{\xi})]$$

for all $\boldsymbol{\lambda}, \boldsymbol{v} \in \mathcal{S}_+^{M-1}$ and $\boldsymbol{\xi} \in \Xi$. By the monotonicity and positive homogeneity of ρ, we get the desired result:

$$\rho[\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[Y_{\boldsymbol{\eta},f}^*(\cdot)]] \max_{m=1,\ldots,M} \frac{\lambda^{(m)}}{v^{(m)}} \geq \rho[\ell_{\boldsymbol{\eta},\boldsymbol{v}}[Y_{\boldsymbol{\eta},f}^*(\cdot)]].$$

Case 2:   Now consider the setting where there exists one direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$ in which the projected length (30) is zero. This implies that $\max_{\mathbf{x}\in\mathbb{X}} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(f(\mathbf{x},\boldsymbol{\xi}))$ is zero on the support of ρ. As the univariate risk functional is partially coherent and the length scalarisation function is strongly monotonic [Tu et al., 2024, Lemma A.2], this implies that all other directions must also have a projected length of zero. Therefore, in this setting, the polar surface (21) reduces down to the degenerate singleton set.

∎

A.7   Proof of Proposition 4.2

Consider the RTS front defined in (22). Firstly, we will show that this set is indeed a polar surface. By definition, to establish that this set is a polar surface, we need to establish that the RTS lengths

$$\sup_{\mathbf{x}\in X} \max_{\mathbf{y}\in\boldsymbol{\rho}[f(\mathbf{x},\cdot)]} s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\mathrm{Len}}(\mathbf{y}) \tag{31}$$

are non-negative and bounded along every positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. This follows simply from the definition of the length scalarisation function (5) and the boundedness of both the objective function and multivariate risk functional. We will now show that the RTS front is either a Pareto front surface or the degenerate singleton set. These two cases are determined by whether the Pareto front surface $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\cup_{\mathbf{x}\in X}\boldsymbol{\rho}[f(\mathbf{x},\cdot)]]$ is non-empty (Case 1) or empty (Case 2), respectively. We will consider both of these separate cases below.

Case 1: Suppose that the Pareto front surface of interest is non-empty. We can then invoke Theorem 4.1 on the Pareto front surfaces $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\cup_{\mathbf{x}\in X}\boldsymbol{\rho}[f(\mathbf{x},\cdot)]]$ and $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\cup_{\mathbf{x}\in X}\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\boldsymbol{\rho}[f(\mathbf{x},\cdot)]]]$. We see very quickly that both of these expressions reduce down to the RTS front (22) and therefore the latter set is indeed a Pareto front surface in this setting.

Case 2: Suppose that the Pareto front surface of interest is empty. This implies that all of the vectors in the set $\cup_{\mathbf{x}\in X}\boldsymbol{\rho}[f(\mathbf{x},\cdot)]$ are weakly dominated by the reference vector. Consequently, this means that all of the RTS lengths (31) are zero, which implies degeneracy: $\mathcal{F}_{\boldsymbol{\eta},f,\boldsymbol{\rho}}^{\text{RTS}}[X] = \{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

■

A.8 Proof of Proposition 4.3

Consider the STR front defined in (23). Firstly, we will show that this set is indeed a polar surface. By definition, to establish that this set is a polar surface, we need to establish that the STR lengths

$$\sup_{\mathbf{x}\in X} \rho[s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\text{Len}}(f(\mathbf{x},\cdot))] \tag{32}$$

are non-negative and bounded along every positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. This follows simply from the definition of the length scalarisation function (5), the boundedness of the objective function and the boundedness of the partially coherent univariate risk functional (Lemma 2.1). We will now show that the STR front is either a Pareto front surface or the degenerate singleton set. These two cases are determined by whether the Pareto front surface $\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[\cup_{\mathbf{x}\in X}\rho[\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[f(\mathbf{x},\cdot)]]]$ is non-empty (Case 1) or empty (Case 2), respectively. We will first recall a useful result before considering both of these separate cases below. Recall that the corresponding risk statistic of the individual Pareto front surface (21) is defined by the set

$$\rho[\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[f(\mathbf{x},\cdot)]] := \{\boldsymbol{\eta} + \rho[s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\text{Len}}(f(\mathbf{x},\cdot))]\boldsymbol{\lambda} \in \mathbb{R}^M : \boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}\} \in \mathbb{L}_{\boldsymbol{\eta}},$$

for every input $\mathbf{x} \in X$. As ρ is a partially coherent univariate risk functional, the following equivalence holds along every positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$:

$$s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\text{Len}}(\rho[\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[f(\mathbf{x},\cdot)]]) = \rho[s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\text{Len}}(f(\mathbf{x},\cdot))]. \tag{33}$$

This equivalence follows directly from the polar parameterised form of the risk statistic and the definition of the length scalarisation function. In essence it says that the point within the risk statistic which achieves the largest projected length along the line $\{\boldsymbol{\eta} + t\boldsymbol{\lambda} \in \mathbb{R}^M : t \in \mathbb{R}\}$ is indeed the vector $\boldsymbol{\eta} + \rho[s_{(\boldsymbol{\eta},\boldsymbol{\lambda})}^{\text{Len}}(f(\mathbf{x},\cdot))]\boldsymbol{\lambda} \in \mathbb{R}^M$.

Case 1: Suppose that the Pareto front surface of interest is non-empty. We can then invoke Theorem 4.1 on this Pareto front surface. By using the equivalence in (33), we see very quickly that this polar representation reduces down to the STR front (23) and therefore the latter set is indeed a valid Pareto front surface in this setting.

Case 2: Suppose that the Pareto front surface of interest is empty. This implies that the sets $\rho[\mathcal{Y}_{\boldsymbol{\eta}}^{\text{int}}[f(\mathbf{x},\cdot)]]$ are degenerate for all inputs $\mathbf{x} \in X$. Consequently, by the equivalence in (33), all of the STR lengths (32) are zero, which implies degeneracy: $\mathcal{F}_{\boldsymbol{\eta},f,\rho}^{\text{STR}}[X] = \{\boldsymbol{\eta}\} \in \mathbb{L}_{\boldsymbol{\eta}}$.

∎

A.9 Proof of Proposition 4.4

Consider any bounded objective function $f : \mathbb{X} \times \Xi \to \mathbb{R}^M$ and strongly dominated reference vector $\boldsymbol{\eta} \in \cap_{\mathbf{x}\in\mathbb{X}} \cap_{\mathbf{y}\in f(\mathbf{x},\Xi)} \mathbb{D}_{\prec}[\{\mathbf{y}\}]$. Let $y_{\mathbf{x},\boldsymbol{\xi}}^{(m)} = f^{(m)}(\mathbf{x},\boldsymbol{\xi})$ denote the m-the objective value at the input $\mathbf{x} \in \mathbb{X}$ and uncertain parameter $\boldsymbol{\xi} \in \Xi$. Then for the upper bounds, we have that

$$\sup_{\boldsymbol{\xi}\in\Xi} \sup_{\mathbf{x}\in\mathbb{X}} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(y_{\mathbf{x},\boldsymbol{\xi}}^{(m)}) = \sup_{\mathbf{x}\in\mathbb{X}} \sup_{\boldsymbol{\xi}\in\Xi} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(y_{\mathbf{x},\boldsymbol{\xi}}^{(m)}) \leq \sup_{\mathbf{x}\in\mathbb{X}} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(\sup_{\boldsymbol{\xi}\in\Xi} y_{\mathbf{x},\boldsymbol{\xi}}^{(m)})$$

for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. Equivalently, this means that $u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{PS}} = u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{STR}} \leq u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{RTS}}$ for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. Similarly, for the lower bounds, we have that

$$\sup_{\mathbf{x}\in\mathbb{X}} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(\inf_{\boldsymbol{\xi}\in\Xi} y_{\mathbf{x},\boldsymbol{\xi}}^{(m)}) = \sup_{\mathbf{x}\in\mathbb{X}} \inf_{\boldsymbol{\xi}\in\Xi} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(y_{\mathbf{x},\boldsymbol{\xi}}^{(m)}) \leq \inf_{\boldsymbol{\xi}\in\Xi} \sup_{\mathbf{x}\in\mathbb{X}} \min_{m=1,\ldots,M} z_{\boldsymbol{\eta},\boldsymbol{\lambda}}^{(m)}(y_{\mathbf{x},\boldsymbol{\xi}}^{(m)})$$

for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. Equivalently, this means that $-l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{PS}} \leq -l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{STR}} = -l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{RTS}}$ for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$. The final result is then obtained by combining these two results together and using the fact that $l_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{PS}} \leq u_{\boldsymbol{\eta},\boldsymbol{\lambda},f}^{\text{PS}}$ for any positive direction $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$.

∎

A.10 Hypervolume of a Pareto front surface

We now recall a useful result in Lemma A.1, which states that the hypervolume of any Pareto front surface can be written as an integral over the set of positive unit vectors. A similar result was presented in the work by Deng and Zhang [2019] and Zhang and Golovin [2020] for the standard hypervolume indicator (29).

Lemma A.1 (Hypervolume of a Pareto front surface) *Consider a reference vector* $\boldsymbol{\eta} \in \mathbb{R}^M$ *and any Pareto front surface* $A \in \mathbb{Y}_{\boldsymbol{\eta}}^*$, *then the hypervolume of this polar surface can be written as*

$$\int_{\mathbb{R}^M} \mathbb{1}[\mathbf{z} \in \mathbb{D}_{\preceq}[A] \cap \mathbb{D}_{\gg}[\{\boldsymbol{\eta}\}]]d\mathbf{z} = \mathbb{E}_{\boldsymbol{\lambda}\sim\text{Uniform}(\mathcal{S}_+^{M-1})}[\tau^{\text{HV}}(\ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[A])].$$

Proof. By rewriting the volume integral in polar coordinates, we obtain the equation

$$\int_{\mathbb{R}^M} \mathbb{1}[\mathbf{z} \in \mathbb{D}_{\preceq}[A] \cap \mathbb{D}_{\gg}[\{\boldsymbol{\eta}\}]]d\mathbf{z} = c_M \int_{\boldsymbol{\lambda}\in\mathcal{S}_+^{M-1}} \ell_{\boldsymbol{\eta},\boldsymbol{\lambda}}[A]^M d\boldsymbol{\phi}(\boldsymbol{\lambda}),$$

where $\boldsymbol{\phi}$ denotes uniform measure over the positive unit vector $\boldsymbol{\lambda} \in \mathcal{S}_+^{M-1}$, whilst $c_M = \int_{\boldsymbol{\lambda}\in\mathcal{S}_+^{M-1}} d\boldsymbol{\phi}(\boldsymbol{\lambda}) = \pi^{M/2}/(2^M \Gamma(M/2 + 1))$ is the volume of the unit hypersphere lying in the positive orthant.

∎

A.11 Proof of Proposition 5.1 and Proposition 5.2

The proof of Proposition 5.1 (or Proposition 5.2) follows directly from the definition of the RTS front (22) (or STR front (23)), together with Proposition 4.2 (or Proposition 4.3), Lemma A.1 and the monotonicity of the hypervolume transformation.

<div align="right">■</div>

B References

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. Cited on pages 7 and 16.

Charles Audet, Jean Bigeon, Dominique Cartier, Sébastien Le Digabel, and Ludovic Salomon. Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292(2):397–422, July 2021. Cited on page 22.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, August 2009. Cited on pages 2, 12, and 32.

Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, January 2011. Cited on pages 2 and 32.

Rasmus Bokrantz and Albin Fredriksson. Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. *European Journal of Operational Research*, 262 (2):682–692, October 2017. Cited on page 9.

Marco Botte and Anita Schöbel. Dominance for multi-objective robust optimization concepts. *European Journal of Operational Research*, 273(2):430–440, March 2019. Cited on page 2.

Carlos A. Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a co-evolutionary multi-objective evolutionary algorithm. In *MICAI 2004: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 688–697. Springer, 2004. Cited on page 23.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200. Curran Associates, Inc., 2021. Cited on page 31.

Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A. Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective Bayesian optimization under input noise. In *International Conference on Machine Learning*, pages 4831–4866. PMLR, June 2022. Cited on pages 15 and 31.

Kalyanmoy Deb and Himanshu Gupta. Searching for robust Pareto-optimal solutions in multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 3410, pages 150–164. Springer Berlin Heidelberg, 2005. Cited on page 8.

Kalyanmoy Deb and Himanshu Jain. Handling many-objective problems using an improved NSGA-II procedure. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012. Cited on pages 2 and 5.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2):182–197, April 2002. Cited on page 31.

Jingda Deng and Qingfu Zhang. Approximating hypervolume and hypervolume contributions using polar coordinate. *IEEE Transactions on Evolutionary Computation*, 23(5):913–918, October 2019. Cited on pages 24 and 36.

Matthias Ehrgott. *Multicriteria Optimization*. Springer-Verlag, 2005. Cited on page 16.

Matthias Ehrgott, Jonas Ide, and Anita Schöbel. Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31, November 2014. Cited on pages 8, 12, and 13.

Michael T. M. Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, August 2006. Cited on pages 29 and 31.

Jörg Fliege and Ralf Werner. Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research*, 234(2):422–433, April 2014. Cited on pages 2, 8, 12, and 13.

Christiane Gerstewitz. Nichtkonvexe Dualität in der Vektoroptimierung. *Wissenschaftliche Zeitschrift der TH LeunaMerseburg*, pages 357–364, 1983. Cited on pages 2 and 15.

Patrick Groetzner and Ralf Werner. Multiobjective optimization under uncertainty: A multi-objective robust (relative) regret approach. *European Journal of Operational Research*, 296 (1):101–115, January 2022. Cited on page 2.

Michael P. Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. *Technical Report, Institute of Mathematical Modeling*, 1998. Cited on pages 3 and 22.

Jonas Ide and Elisabeth Köbis. Concepts of efficiency for uncertain multi-objective optimization problems based on set order relations. *Mathematical Methods of Operations Research*, 80(1): 99–127, August 2014. Cited on pages 2, 12, and 13.

Jonas Ide and Anita Schöbel. Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271, January 2016. Cited on page 2.

Jonas Ide, Elisabeth Köbis, Daishi Kuroiwa, Anita Schöbel, and Christiane Tammer. The relationship between multi-objective robustness concepts and set-valued optimization. *Fixed Point Theory and Applications*, 2014(1):83, March 2014. Cited on page 12.

Hisao Ishibuchi, Noritaka Tsukamoto, Yuji Sakane, and Yusuke Nojima. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In *2009 IEEE Congress on Evolutionary Computation*, pages 530–537, May 2009. Cited on pages 2 and 5.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 110–125. Springer International Publishing, 2015. Cited on page 24.

Akhtar A. Khan, Christiane Tammer, and Constantin Zălinescu. *Set-Valued Optimization: An Introduction with Applications*. Springer, 2015. Cited on page 12.

Kathrin Klamroth, Elisabeth Köbis, Anita Schöbel, and Christiane Tammer. A unified approach for different concepts of robustness and stochastic programming via non-linear scalarizing functionals. *Optimization*, 62(5):649–671, May 2013. Cited on pages 15 and 16.

Kathrin Klamroth, Elisabeth Köbis, Anita Schöbel, and Christiane Tammer. A unified approach to uncertain optimization. *European Journal of Operational Research*, 260(2):403–420, July 2017. Cited on page 15.

Joshua Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006. Cited on page 31.

Daishi Kuroiwa and Gue Lee. On robust multiobjective optimization. *Vietnam Journal of Mathematics*, 40, January 2012. Cited on pages 8 and 12.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer US, 1998. Cited on pages 5 and 33.

Seyedali Mirjalili and Andrew Lewis. Novel performance metrics for robust multi-objective optimization algorithms. *Swarm and Evolutionary Computation*, 21:1–23, April 2015. Cited on page 22.

Adriano Pascoletti and Paolo Serafini. Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42(4):499–524, April 1984. Cited on page 16.

András Prékopa. Multivariate value at risk and related topics. *Annals of Operations Research*, 193(1):49–69, March 2012. Cited on pages 8 and 9.

Anna Grazia Quaranta and Alberto Zaffaroni. Robust optimization of conditional value at risk and portfolio selection. *Journal of Banking & Finance*, 32(10):2046–2056, October 2008. Cited on pages 2 and 16.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv.1908.05659*, August 2019. Cited on pages 2, 7, and 32.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006. Cited on page 27.

Ralph Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, July 2002. Cited on page 7.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48: 67–113, October 2013. Cited on pages 16 and 17.

Ludger Rüschendorf. *Mathematical Risk Analysis: Dependence, Risk Bounds, Optimal Allocations and Portfolios*. Springer Series in Operations Research and Financial Engineering. Springer, 2013. Cited on page 6.

Marie Schmidt, Anita Schöbel, and Lisa Thom. Min-ordering and max-ordering scalarization methods for multi-objective robust optimization. *European Journal of Operational Research*, 275(2):446–459, June 2019. Cited on page 13.

Katrin Schöttle and Ralf Werner. Robustness properties of mean-variance portfolios. *Optimization*, 58(6):641–663, August 2009. Cited on page 13.

Oliver Schutze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, August 2012. Cited on page 23.

Ke Shang, Hisao Ishibuchi, Min-Ling Zhang, and Yiping Liu. A new R2 indicator for better hypervolume approximation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pages 745–752. Association for Computing Machinery, July 2018. Cited on page 24.

Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, January 2017. Cited on page 7.

Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective Bayesian optimization using Pareto-frontier entropy. In *International Conference on Machine Learning*, volume 37, pages 9279–9288. PMLR, November 2020. Cited on page 31.

Christiane Tammer and Petra Weidner. *Scalarization and Separation by Translation Invariant Functions*. Springer, June 2020. Cited on pages 2, 15, and 16.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89, April 2020. Cited on page 28.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 9922–9938. Curran Associates, Inc., 2022. Cited on page 31.

Ben Tu, Nikolas Kantas, Robert M. Lee, and Behrang Shafei. Random Pareto front surfaces. *arXiv:2405.01404*, June 2024. Cited on pages 2, 3, 5, 17, 18, 19, 33, and 34.

Ben Tu, Nikolas Kantas, Robert M. Lee, and Behrang Shafei. Multiobjective optimization using the R2 utility. *SIAM Review*, 67(2):213–255, May 2025. Cited on pages 4, 19, 22, and 24.

Rajkumar Vaidyanathan, Kevin Tucker, Nilay Papila, and Wei Shyy. CFD-based design optimization for single element rocket injector. In *41st Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings, pages 1–21. American Institute of Aeronautics and Astronautics, January 2003. Cited on page 28.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, volume 37, pages 11096–11105. PMLR, November 2020. Cited on pages 24 and 36.

Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - A comparative case study. In *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, pages 292–301. Springer, 1998. Cited on page 24.