Horsetail Matching: A novel approach to robust optimization under probabilistic and mixed uncertainties.

L W Cook

Abstract

Keywords: , Robust optimization, Optimization under uncertainty, Mixed uncertainties, Interval uncertainties, Horsetail matching

1. Introduction

The design of real engineering systems is subject to many uncertainties, and the engineering community has realized the importance of including these in the design process, leading to the development of myriad robust optimization techniques, where the goal is to obtain a design which will both perform well and be insensitive to variation due to these uncertainties [1].

The focus historically has been on using probabilistic techniques to model uncertainty and optimizing properties of the output probability distribution (statistical moments for example). However, different types of uncertainties exist and input probability distributions are not always available; in recent years the importance of distinguishing between aleatory and epistemic uncertainties has been recognized, leading to the development of effective methods of mixed uncertainty propagation. However, robust optimization under mixed uncertainties has not received much attention in the literature and current approaches are limited.

This paper proposes a novel formulation of robust optimization under mixed uncertainties referred to as horsetail plot matching. It is designed to overcome some of the limitations of existing techniques for robust optimization, under probabilistic uncertainties as well as under mixed interval-probabilistic uncertainties.

1.1. Background

A traditional optimization considers an objective function q, of a system (e.g. efficiency, cost, weight etc) as a function of design variables $\boldsymbol{x}$ which can be modified, and the aim to find a design $\boldsymbol{x}^*$ such that:

$$q(\boldsymbol{x}^*) \leq q(\boldsymbol{x}) \ \forall \boldsymbol{x} \in \mathcal{X} \tag{1}$$

$$\text{s.t. } x_l < x_i < x_u \ \forall i = 1, ..., n_{dv} \tag{2}$$

$$g_j(\boldsymbol{x}) \leq 0 \ \forall j = 1, ..., n_c \tag{3}$$

However, in reality, many uncertainties, $\boldsymbol{u}$, exist in the system (from a variety of sources e.g. CITATION) and so the objective function becomes $q(\boldsymbol{x}, \boldsymbol{u})$. Ideally a global solution is found such that:

$$q(\boldsymbol{x}^*, \boldsymbol{u}) \leq q(\boldsymbol{x}, \boldsymbol{u}) \ \ \forall \boldsymbol{x} \in \mathcal{X}, \ \forall \boldsymbol{u} \in \mathcal{U} \tag{4}$$

Methods for optimization under uncertainty are usually classified as robust design optimization or reliability based design optimization CITATION. Robust design optimization considers the influence of uncertainty on the objective function and optimizes a system to both perform well and be resistant to uncertainties. Reliability based design on the other hand concerns the influence of uncertainties on the constraints. This is more important for constraints where violation of them results in failure of the system (e.g. stress not exceeding the critical value).

There has been more work into RBDO under mixed uncertainties, since what to do with the horsetail plot becomes more obvious.

List types of approaches to robust optimization: focus on probability but mention that there are very limited techniques available for mixed uncertainties.

Optimization under uncertainty is split into RDO and RBDO - the difference being on whether the uncertainty is on the objective function(s) or constraints. In real problems there are often many things which could be taken as objective functions or constraints

1.2. Limitations of existing approaches

Discuss: my philosophy about robust optimization. We are trying to “maximize the chance of the objective function being as small as possible”. This means that treating mean and variance as two independent objectives is not that appropriate, and this is my main issue with weighted sum or Pareto front approaches: for objective functions where smaller/larger is always better (e.g. efficiency, cost, weight), there is only a certain penalty to the mean we would be willing to accept. It is possible that a designer may value robustness very highly (a nominal is better type scenario maybe), but in this case then this variable might be better off as a constraint, or if not then horsetail matching can deal with nominal is better very well with using the target. Finding the entire Pareto front is therefore unnecessary a lot of the time and wastes computational resources. The weighted sum has similar limitations: we could end up anywhere on the Pareto front and from my experience usually end up with a conservative solution. This is part of the inspiration for density matching: a single objective formulation that deals directly with the PDF instead of mean/variance. However, it is not straightforward to define what makes an objectively better robust solution when dealing with PDFs, but it does with CDFS: non-overlapping. Therefore we can define "objectively worse" CDFs from a robust optimization point of view. This is part of the motivation for horsetail matching.

Discuss: limited approaches to mixed uncertainties: the three weighted sum or three Pareto front are very cumbersome and suffer the limitations already discussed even further.

2. Horsetail Matching Approach

2.1. Description

Describe in terms of the mixed interval + probabilistic horsetail plot, and then limit it to just probabilistic uncertainties for now.

The concept is to minimize the distance between the horsetail plot of the current design and that of a target, as illustrated in figure 1.

Figure 1: Illustration of the horsetail matching concept

We assume the function $q = q(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{e})$ is a continuous, scalar valued function of the design variables $\boldsymbol{x}$, aleatory uncertainties $\boldsymbol{a}$, epistemic uncertainties $\boldsymbol{e}$. We assume that the aleatory uncertainties are independent and each is defined on a probability space with a given probability density function and hence cumulative density function. We assume that the epistemic uncertainties are independent and each is defined by an interval $[l_i, u_i]$. At a fixed design point $\boldsymbol{x}_0$, let $q_x = q(\boldsymbol{x}_0, \boldsymbol{a}, \boldsymbol{e})$, let the upper bound of the possible

CDFs be given by the function $F_1(q_x)$: $\mathbb{R} \rightarrow [0,1]$ and similarly the lower bound by $F_2(q_x)$: $\mathbb{R} \rightarrow [0,1]$; these two curves define the horsetail plot and vary as the design variables in $\boldsymbol{x}$ change.

The target horsetail plot is given by two curves: T_1 and T_2: $\mathbb{R} \rightarrow [0,1]$, which represent the desired shape of the horsetail plot: the desired behavior of the quantity of interest under both epistemic and aleatory uncertainties. Also define $F_i^{-1}(h)$ and $T_i^{-1}(h)$: $[0,1] \rightarrow \mathbb{R}$ as the inverse of the curves (since by definition the curves are weakly monotonically increasing, the inverse will exist). In this work we propose the following L_p-norm of the difference in q-space of the horsetail plots as a distance metric between each horsetail curve of a given design and the target:

$$d_p(\boldsymbol{x}) = \Big(\int_0^1 \ \big(\ F_1^{-1}(h) - T_1^{-1}(h) \ \big)^p \ dh + \int_0^1 \ \big(\ F_2^{-1}(h) - T_2^{-1}(h) \ \big)^p \ dh \ \Big)^{1/p} \tag{5}$$

Where h is a variable in horsetail space integrated between 0 and 1, and p is a power $\in (1, \infty]$. Then the optimization problem becomes finding $\boldsymbol{x}^*$ such that:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmin}} \ d_p(T_1, T_2, F_1(q_x), F_2(q_x)) \tag{6}$$

Where the value x^* corresponds to the optimal design under uncertainty: its behavior under uncertainty is as close as possible to that specified in the target.

Under probabilistic uncertainty:

$$d_p(\boldsymbol{x}) = \Big(\ 2 \int_0^1 \ \big(\ F^{-1}(h) - T^{-1}(h) \ \big)^p \ dh \ \Big)^{1/p} \tag{7}$$

In the interval case, the CDF curves reduce to step functions at the minimum or maximum value of f over the uncertainty and the metric becomes:

$$d_p(\boldsymbol{x}) = \Big(\int_0^1 \ \big(\ f_{min} - T_1^{-1}(h) \ \big)^p \ dh + \int_0^1 \ \big(\ f_{max} - T_2^{-1}(h) \ \big)^p \ dh \ \Big)^{1/p} \tag{8}$$

2.2. Discussion

Discuss the influence of p - in the situation where no target is provided and we are just interested in the minimization problem, p allows the method to reduce to existing techniques.

IF ROOM: how the distance also influences the trade-off between performance and robustness, but how this is much smaller compared to the actual trade-off and the value of p.

2.3. Numerical Implementation

Explain why p = 2 chosen: we do not want p = 1 since then we are only using the mean and not taking into account any measure of variance, therefore still going with our philosophy of "maximizing the chance of being as good as possible" we choose p = 2 so that it takes into account variance but still following our philosophy.

2.3.1. Evaluating the Integral

Furthermore, using the l-2 norm makes numerical implementation much nicer with trapezium rule integration and gradients.

For each individual horsetail curve, we must evaluate the L_2 norm using numerical integration. For each CDF, we can express the L_2 norm as:

$$\hat{D}_i = \sum_{i=1}^{N} w_i \big(t(h_i) - q_{\boldsymbol{x}}(h_i)\big)^2 \tag{9}$$

Various numerical integration approaches exist, however for this formulation we use the trapezium rule since it is much more straightforward to find an expression/approximation for the CDF as opposed to the inverse.

T^{-1} should be known since the target distribution is specified, but F^{-1} is not available directly from most uncertainty propagation techniques. However, this objective function can be evaluated without access to F^{-1} itself (as long as the function $F(q)$ or an approximation of it is known) as follows: choose samples of the objective function q to give q_i $(i = 1, ..., N)$ and obtain the corresponding $F(q)$ so we have N pairs of (q_i, h_i). Using the known inverse of the target distribution T^{-1}, evaluate the value of t_i (in objective function space) corresponding to the evaluated values of $h_i = F(q_i)$ to obtain N pairs of (t_i, h_i). We now have everything required for a trapezium rule integration of (9). The trapezium rule integration is necessary since the spacing of the points h_i is determined by the CDF of the current design $F(q)$ and so we cannot control it. However using a large N we can still obtain good accuracy of integration. This method is illustrated in figure 2.

Where $q(h_i)$ / $t(h_i)$ is the value in objective function space given by $F^{-1}(h_i)$ / $T^{-1}(h_i)$, and w_i is the quadrature weighting corresponding to the quadrature point h_i.

Figure 2: Illustration of the numerical integration used to estimate the objective function

Under this trapezium rule, we can express the integral as a matrix multiplication as follows:

$$\hat{D}_i = (\boldsymbol{t} - \boldsymbol{q})^T \boldsymbol{W} (\boldsymbol{t} - \boldsymbol{q}) \tag{10}$$

Where $\boldsymbol{t}_i = t(h_i)$, $\boldsymbol{q}_i = q(h_o)$, and the weighting matrix W is a diagonal matrix with entries given by:

$$\begin{align} W_{1,1} &= 0.5(\ h_2 - h_1\ ) \tag{11} \\ W_{k,k} &= 0.5(\ h_{k+1} - h_{k-1}\ )\ \forall k = 2, ..., N-1 \tag{12} \\ W_{N,N} &= 0.5(\ h_N - h_{N-1}\ ) \tag{13} \end{align}$$

Other quadrature rules which require certain spacing of h_i would not be possible without being able to estimate F^{-1} directly.

Because we are able to write the objective in terms of matrices, we can compute the gradient as follows:

Note that the vector $\boldsymbol{q}$ is fixed, since it is the points used for the trapezium rule. The weighting matrix $\boldsymbol{W}$ is a diagonal matrix with values of h along the diagonal, and so is a function of the design, and the vector $\boldsymbol{t}$ depends on the values of h and so is also a function of the design.

$$\frac{d\hat{D}}{dx_k} = 2(\boldsymbol{q} - \boldsymbol{t})^T \boldsymbol{W} \frac{\partial \boldsymbol{t}}{\partial x_k}\ +\ (\boldsymbol{q} - \boldsymbol{t})^T \frac{\partial \boldsymbol{W}}{\partial x_k} (\boldsymbol{q} - \boldsymbol{t}) \tag{14}$$

Where $\frac{\partial W}{\partial x_k}$ is a diagonal matrix.

This requires $\big(T^{-1}(h_i)\big)'$, which is the derivative of the target inverse CDF with respect to the argument, which could also be supplied analytically along with the target, or can easily be found via finite differencing.

$$\begin{align} (\frac{\partial \boldsymbol{W}}{\partial x_k})_{i,i} &= \frac{\partial W_{i,i}}{\partial x_k} \tag{15} \\ (\frac{\partial \boldsymbol{t}}{\partial x_k})_i &= \frac{\partial t_i}{\partial x_k} = \big(T^{-1}(h_i)\big)' \frac{\partial h_i}{\partial x_k} \tag{16} \end{align}$$

This can be shown by considering and differentiating each term of the trapezium rule summation for D individually.

2.3.2. Evaluating the horsetail curves

Now we turn our attention to evaluating h = F(q) for use in this implementation.

In many practical applications, a (non-linear) simulation will be used to evaluate the objective function, so an exact form of the CDF of an output under probabilistic uncertainties will not be available. Therefore we need a method of estimating the CDF.

kernel density estimation can be used to approximate both CDFs [2]: using an M-point kernel density approximation to the CDF by sampling $q(\boldsymbol{x}_0, \boldsymbol{a}, \boldsymbol{e}_{ext})$ at M realizations of the aleatory uncertainty to obtain values of $q_j = q(\boldsymbol{x}_0, \boldsymbol{a}_j, \boldsymbol{e}_{ext})$, $j = 1, ..., M$, we obtain:

$$F(q) \simeq \hat{F}(q) = \frac{1}{M} \sum_{j=1}^{M} \Phi(q - q_j) \tag{17}$$

Where $\Phi(r) = \int_{-\infty}^{r} K(r')\ dr'$, for kernel function K. For example using a gaussian kernel with bandwidth parameter b:

$$K(r) = \frac{1}{\sqrt{2\pi b}} \exp\big(- \frac{r^2}{2b^2} \big) \tag{18}$$

Then $\Phi(r)$ becomes the error function for this gaussian:

$$\Phi(r) = \frac{1}{b} \operatorname{erf}\big(\frac{r}{b} \big) \tag{19}$$

This kernel density estimation is used to find the N values of h_i in (9), which are used in the matrix W as given by (11).

Similarly using a kernel density estimation for $h = F(q)$, we can obtain the gradients required $\frac{\partial h}{\partial x_k}$:

$$\frac{\partial h}{\partial x_k} \simeq \frac{\partial \hat{F}(q)}{\partial x_k} = \frac{1}{M} \sum_{j=1}^{M} K(q - q_j)\ (-1)\ \frac{\partial q_j}{\partial x_k} \tag{20}$$

Where K is the derivative of Φ: the regular kernel function since Φ was the integral of the kernel function. Therefore the only thing required to work out the gradient is $\frac{\partial q_j}{\partial x_k}$, which is the gradient of the underlying objective function with respect to design variables at a given value of the uncertainties u_j.

If we define $\frac{\partial \boldsymbol{t}}{\partial x_k} = T'\ h'$ and $\boldsymbol{W}' = \frac{\partial \boldsymbol{W}}{\partial x_k}$, then:

$$\frac{d\hat{D}}{dx_k} = 2(\boldsymbol{q}-\boldsymbol{t})^T\boldsymbol{W}\boldsymbol{T}'\boldsymbol{h}' \; + \; (\boldsymbol{q}-\boldsymbol{t})^T\boldsymbol{W}'(\boldsymbol{q}-\boldsymbol{t}) \tag{21}$$

NOTE: for accuracy of this gradient, it is much more important to have a large number of samples to get a good PDF than it is to have a large number of integration points. About 100 integration points from 0 to 1 seems fine.

The advantage of being able to evaluate the gradient like this is that any existing simulation/modelling software/technique for a particular application that has gradient functionality can take advantage of this within the horsetail matching formulation (like density matching), for example when CFD software has adjoint capabilities.

2.3.3. Response Surface Sampling

To get an accurate CDF to use in the formulation, a large number of samples of q_j nee to be obtained. However, when the simulation is computationally expensive, this is infeasible. Therefore a response surface to the objective function over the uncertain space can be created using a small number of samples, and this response surface can then be sampled a large number of times in order to get an accurate kernel density estimation.

In this paper we use simple polynomial response surfaces, but any response surface can be substituted and the technique would still be fine.

3. Application to Probabilistic Uncertainties

TEST PROBLEM for gradient based DM + WS stuff TASOPT for MO and weighted sum stuff.

3.1. Comparison with MO optimization and weighted sum

3.2. Comparison with Density Matching

Computational performance: same if not better - no need for two step approach.

Do not need to supply a target: still works well with a simple unachievable step target.

Intrinsically prefers robust solutions more than DM: DM punishes peaks, HM punishes tails.

4. Application to Mixed Uncertainties

TASOPT - 3D Pareto front with a few different targets.

[1] H.-G. Beyer, B. Sendhoff, Robust Optimization - A Comprehensive Survey, Computer Methods in Applied Mechanics and Engineering 196 (33-34) (2007) 3190–3218. doi:10.1016/j.cma.2007.03.003.
URL `http://linkinghub.elsevier.com/retrieve/pii/S0045782507001259`

[2] D. W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, John Wiley & Sons, Inc., 1992.