Surrogate-based Maximization of Belief Function for Robust Design Optimization

Simone Alicino * and Massimiliano Vasile †

University of Strathclyde, Glasgow, G4 0LT, United Kingdom

This paper proposes an approach based on surrogate models to reduce the computational cost of evidence-based robust design optimization. Evidence Theory provides two quantitative measures, Belief and Plausibility, that defines the lower and upper probability that a given proposition is true under uncertainty. The maximization of the Belief is of great interest to the designers because it provides the design solution such that a given proposition on the system budgets is always true, given the current evidence on the set of uncertain design parameters. The paper introduces a novel min-max multi-objective optimization algorithm to maximize the Belief in multiple conflicting propositions. Then an approach based on surrogate models is presented to substantially reduce the computational cost associated with the optimization of the design solutions that maximize the Belief in the given proposition. A simple test case of spacecraft system design is presented will illustrate how to apply the proposed approach.

Nomenclature

Δt_a	access time, s
Δt_{aq}	acquisition time, s
η_{ant}	antenna efficiency
θ	focal element
ν	generic threshold
ρ_{CMR}	amplifier case mass fraction
ρ_A	antenna specific density, kg/m^2
σ	width parameter
ϕ_f	Faraday rotation, rad
A	generic proposition
A_g	global archive
A_L	atmospheric losses, dB
AM_L	antenna misalignment loss, dB
AMP_T	amplifier noise temperature, K
AN_T	antenna noise temperature, dB
B	on-board data volume, bits
Bel	Belief function
bpa	basic probability assignment
c	speed of light, m/s
D_{ant}	antenna diameter, m
D	design space
$\mathbf{d}$	design parameter vector
e	elevation angle, rad
F_L	feeder loss, dB
F_{SL}	free space loss, dB
F_T	transmitter noise figure

*Ph.D Student, Department of Mechanical and Aerospace Engineering, 141 St. James Road.
†Reader, Department of Mechanical and Aerospace Engineering, 141 St. James Road.

f_T	carrier frequency, MHz
f, g	generic functions
G_T	low noise amplifier gain, dB
G_r	ground station antenna gain, dB
G_t	antenna gain, dB
h	activation function
h_{GS}	ground station altitude, m
I_L	implementation loss, dB
k	Boltzmann constant, dB
L_T	transmitter cable loss, dB
L_{total}	total signal loss, dB
M_{TTC}	communication system mass, kg
M_{ant}	antenna mass, kg
M_{amp}	amplifier mass, kg
M_{case}	casing mass, kg
N_{rain}	rain noise, dB
P_L	polarization mismatch, dB
P_{TTC}	required communication power, dB
Pl	Plausibility function
$\mathbf{R}, \mathbf{r}$	correlation matrix
R_t	required data rate, bits/s
Ra_L	rain absorption loss, dB
RN_{fig}	receiver noise figure
r_{GS}	distance from the ground station, km
S_L	horn antenna lateral surface, m^2
S_{temp}	transmitter noise temperature, K
T_{data}	transmitted data, bits
TS_{noise}	total system noise, dB
U	uncertain space
$\mathbf{u}$	uncertain parameter vector
w	radial basis network weight

I. Introduction

The optimization of the design of an engineering system may lead to solutions that, though optimal with respect to one or more criteria, are highly sensitive to uncertainty in some of the design parameters. In this sense these solution are not robust as the value of the design criteria is subject to significant variations under uncertainty. In the preliminary design of an engineering system or component the type of uncertainty is generally epistemic as more information is available as the design process progresses. In order to overcome this initial lack of information and capture the effect of epistemic uncertainty into the design process, traditional approaches add margins to the system budgets (e.g. mass and power). Although adding margins is a quick and consolidated approach it does not rely on a quantitative measure of the impact of uncertainty on the design budgets and can lead to an underestimation of the impact or to an overestimation of the design budget (e.g. an excessive system mass).[1] As demonstrated by authors as Oberkampf et al.[2] Evidence Theory offers an interesting way to model both epistemic and aleatory uncertainties in the design of engineering systems. Vasile[3] and Croisard et al.[4] provided some examples of application of Evidence Theory to the robust optimal design of space systems and space trajectories. The benefit coming from the correct modelling of epistemic uncertainty is considerable as it provides a rigorous quantification of the system budgets, but the computation of Belief and Plausibility requires running a number of optimizations that can grow exponentially with the number of dimensions and can become intractable even for problems of moderate size. Recently, Vasile et al.[1] proposed some strategies to obtain an estimation of Belief and Plausibility with a substantial reduction of the computational cost. The approach in Vasile et al.[1] requires, as a first step, the solution of a single-objective, or multi-objective, min-max problem. The solution of this min-max problem corresponds to a 'worst case scenario' type of design in which, for example, the design budgets are minimized in the worst

case. Although this first step avoids the intrinsic exponential complexity in the computation of Belief and Plausibility, nonetheless it requires a high number of function evaluations that might be prohibitive if the system model is computationally intensive. This paper presents an approach based on surrogate models to reduce the computational cost associated to the solution of the worst case scenario design problem. The paper starts with a brief introduction to Evidence Theory and its use in the context of robust design optimization. It then introduces an algorithm to compute a multi-objective optimal design solution under uncertainty. The use of surrogate models to reduce the computational cost in single and multi-objective optimization cases is then presented. The preliminary robust 'worst case' design of the telecommand and telemetry system of a satellite is finally used to illustrate the application of Evidence-based Robust Design Optimization to the design of space systems.

II. Evidence-Based Robust Design Optimization

Evidence Theory, introduced by Shafer in 1976,[5] was conceived to adequately model both epistemic and aleatory uncertainty when no information on the probability distributions is available. Furthermore, the theory provides a nice framework to incorporate multiple pieces of evidence in support to a statement, or proposition. During the preliminary design of an engineering system, experts can provide informed opinions by expressing their belief in an uncertain parameter u being within a certain set of intervals. The level of confidence an expert has in one of the intervals u can belong to is quantified by using a Basic Probability Assignment (*bpa*). Note that the *bpa* is actually a belief rather than an actual probability. All the intervals form the so-called frame of discernment Θ, which is a set of mutually exclusive elementary propositions. The frame of discernment can be viewed as the counterpart of the finite sample space in probability theory. The power set of Θ is called $U = 2^{\Theta}$ or the set of all the subsets of Θ (the uncertain space in the following). An element θ of U that has a non-zero *bpa* is named a focal element. When more than one parameter is uncertain, the focal elements are the result of the Cartesian product of all the elements of each power set associated to each uncertain parameter. The *bpa* of a given focal element is then the product of the *bpa* of all the elements in the power set associated to each parameter. All the pieces of evidence completely in support of a given proposition form the cumulative belief function Bel, whereas all the pieces of evidence partially in support of a given proposition form the cumulative plausibility function Pl. For example, a proposition can be that a given design budget assumes values that are below a given threshold ν under uncertainty. In mathematical form this can be expressed as:

$$A = \{\mathbf{u} \in U | f(\mathbf{u}) \leq \nu\} \tag{1}$$

where A is the proposition about which the Belief and Plausibility need to be evaluated, f is the outcome of the system model and the threshold ν is the desired value of a design budget (e.g. the mass). It is important to note that the set A can be disconnected or present holes, likewise the focal elements can be disconnected or partially overlapping.

II.A. Multi-Objective Robust Design Formulation

Let us consider a function $f : D \times U \subseteq \Re^{m+n} \rightarrow \Re$ characterizing an engineering system to be optimized, where D is the available design space and U the uncertain space. The function f represents the model of the system budgets (e.g. power budget, mass budget, etc.), and depends on some uncertain parameters $\mathbf{u} \in U$ and design parameters $\mathbf{d} \in D$. What designers are usually interested in is the maximum variation of the function f with u, in other words they are interested in the variation of the optimal Belief with the threshold ν. If m objective functions exist, then the following two problems can be solved without considering all the focal elements:

$$\boldsymbol{\nu}_{\min} = \min_{\mathbf{d} \in D} [\min_{\mathbf{u} \in \bar{U}} f_1(\mathbf{d}, \mathbf{u}), \min_{\mathbf{u} \in \bar{U}} f_2(\mathbf{d}, \mathbf{u}), \ldots, \min_{\mathbf{u} \in \bar{U}} f_m(\mathbf{d}, \mathbf{u})] \tag{2}$$

$$\boldsymbol{\nu}_{\max} = \min_{\mathbf{d} \in D} [\max_{\mathbf{u} \in \bar{U}} f_1(\mathbf{d}, \mathbf{u}), \max_{\mathbf{u} \in \bar{U}} f_2(\mathbf{d}, \mathbf{u}), \ldots, \max_{\mathbf{u} \in \bar{U}} f_m(\mathbf{d}, \mathbf{u})] \tag{3}$$

Problem 2 is a multiobjective minimization over the Cartesian product of the uncertain and design space $\bar{U} \times D$, where $\bar{U}$ is a unit hypercube collecting all the focal elements in a compact set with no overlapping or holes. Problem 3 looks for the minimum of the maxima of all the functions over $\bar{U}$ and represents the 'worst case scenario' design, the solution of which is the main topic of this paper. Note that the

Algorithm 1 Main MACSν Algorithm

```
while n_feval < n_feval,max do
    Run individualistic moves and generate candidate population P_i and trial samples P_t
    Run Algorithm 2: Minmax Selection
    if P_i ∨ P_t ≻ A_k or P_i ∨ P_t ≺ A_k then
        Run Algorithm 3: Cross-check
    end if
    Update P_k and A_k
    Run social moves and generate candidate population P_s
    Run Algorithm 2: Minmax Selection
    if P_s ≻ A_k or P_s ≺ A_k or ‖P_s − A_k‖ > ε then
        Run Algorithm 3: Cross-check
    end if
    Update P_k and A_k
    Run Algorithm 4: Validation
    if P_k ≻ A_k or P_k ≺ A_k then
        Run Algorithm 3: Cross-check
    else if ‖P_k − A_k‖ < δ then
        Replace u ∈ P_k with u ∈ A_k
    end if
end while
```

maximum of every function is independent of the other functions and corresponds to a different uncertain vector. Therefore, all the maxima can be computed in parallel with m single objective maximizations. The maximization of each function is performed by running a global optimization over $\bar{U}$ using the version of Inflationary Differential Evolution implemented in IDEA.[6] The minimization over D is performed with a modified version of MACS2,[7] called MACSν (see Algorithm 0), which implements some special heuristics to increase the probability that each maximization identifies the global maximum. In particular, once a new candidate population is generated after either individualistic or social moves, Algorithm 2 is run to select the design vectors to attribute to the next generation. Algorithm 2 implements the following heuristics: if the **d** vector is unchanged, the old **u** vector is replaced with the new one, if the new one yields a higher value of the objective function; if the **d** vector has changed and the **u** vector has not, the new **d** vector replaces the old one if it reduces the value of the objective function; if both the **d** and the **u** vector are different the new vectors replace the old ones. At the end of this selection, Algorithm 3 is used to discriminate and archive the solutions that are Pareto dominant. Algorithm 3 performs a cross check that is necessary to be able to compare the values of the objective functions for a newly generated design vector against the function values computed of an already archived solution. Algorithm 3 scans through all the design vectors $\mathbf{d}_{i,k}$ in the population P_k at iteration k of MACSν with $i \in \{1, \ldots, n_{pop}\}$ and for each objective function f_l, if the current uncertain vector $\mathbf{u}_{i,k}$ differs from the archived one $\mathbf{u}^l_{i,arch}$, compares the archived objective function to the objective function associated to the archived design vector $\mathbf{d}_{i,arch}$ and to the local maximum $\tilde{\mathbf{u}}^l_i$ computed running a local maximization from $\mathbf{u}^l_{i,k}$. If the new objective is better then the archived one, then it and its uncertain vector $\tilde{\mathbf{u}}^l_i$ replace the archived ones. Consider, in fact, that a different design vector can correspond to a different landscape and therefore to a different location of the maxima. Algorithm 3 also compares the current objective function $f_l(\mathbf{d}_{i,k}, \mathbf{u}^l_{i,k})$ to the objective function associated to the current design vector $\mathbf{d}_{i,k}$ and to the local maximum $\tilde{\mathbf{u}}^l_{i,arch}$ computed running a local maximization from $\mathbf{u}^l_{i,arch}$. If the new objective is better then the current one, then it and its uncertain vector $\tilde{\mathbf{u}}^l_{i,arch}$ replace the current ones. After the individualistic and social moves have been performed, Algorithm 4 is run. Algorithm 4 runs a global optimization with probability p_r, or a local optimization with probability $1 - p_r$ for the archived population until there is no variation in the archived objective functions.

Algorithm 2 Minmax Selection

```
for all d_{i,k} ∈ P_k do
    if ‖d_{i,new} − d_{i,k}‖ = 0 then
        for all l ∈ {1, ..., m} do
            if f_l(d_{i,new}, u^l_{i,new}) ≥ f_l(d_{i,k}, u^l_{i,k}) then
                replace u^l_{i,k} with u^l_{i,new}
            end if
        end for
    else if ‖d_{i,new} − d_{i,k}‖ > 0 then
        for all l ∈ {1, ..., m} do
            if ‖u^l_{i,new} − u^l_{i,k}‖ > 0 then
                replace u^l_{i,k} with u^l_{i,new}
                replace d_{i,k} with d_{i,new}
            else
                if f_l(d_{i,new}, u^l_{i,new}) < f_l(d_{i,k}, u^l_{i,k}) then
                    replace d_{i,k} with d_{i,new}
                end if
            end if
        end for
    end if
end for
```

III. Surrogate Modelling

Current developments[1] have shown that the solution of the minmax problem, involving multi-modal systems with up to 80 design parameters, can be treated on a standard machine in tens of minutes of computational time. In this context, the surrogate-based approach can play a valuable role. The surrogates are constructed using data drawn from high-fidelity models, and provide fast approximations of the objectives and constraints at new design points. An overview of surrogate-based analysis and optimization can be found in Queipo et al.[8] In this paper we have used Kriging predictor and Radial Basis Function (RBF) Networks as surrogate models. The Kriging predictor estimates the response of a function at some unsampled location as the sum of two components: a regression model (e.g., polynomial trend) and a correlation model representing the fluctuations around the trend, with the basic assumption that these are correlated and the correlation depends only on the distance between the locations under consideration. Moreover, the Kriging predictor yields an estimate of the prediction error, in the form of the mean square error. This feature is very convenient since the prediction error can be exploited in the surrogate update strategy. Radial Basis Function Networks are a special type of artificial neural networks in which radial basis functions are used as activation function. For this reason, RBF networks have the feature of being very good at interpolation. Moreover, their learning phase is shorter, i.e. they learn faster, than common artificial neural networks. However, unlike the Kriging predictor, they do not provide an estimate of the prediction error, therefore an approach, based on the minimum distance in the design/uncertain space, has been devised for the surrogate update strategy.

III.A. Kriging Predictor

Starting from a set of sample points, or design sites, the Kriging predictor is an interpolation technique that makes use of a regression function and a correlation model to predict the response of a function at a desired point. Being an interpolation method, it gives an exact prediction of the response at the sample points. Moreover, it assumes that the output function values are correlated in design space, i.e. closer points are more highly correlated. A complete derivation of the Kriging model can be found in Jones.[9] If we suppose to have a set of n design sites $(\mathbf{x}, \mathbf{y})$, the correlation matrix $\mathbf{R}$ of the design points can be expressed as

$$\mathbf{R} = [R_{ij}] = \exp\left[-\sum_{l=1}^{d} \theta_l \|\mathbf{x}_{il} - \mathbf{x}_{jl}\|^{p_l}\right] \tag{4}$$

Algorithm 3 Cross-check

for all $\mathbf{d}_{i,k} \in P_k$ **do**
  Take the function values $\mathbf{f}_{arch}$ with associated design vector $\mathbf{d}_{i,arch}$ and uncertain matrix $\mathbf{u}_{i,arch} \in$ the current archive A_k
  for all $l \in \{1, \ldots, m\}$ **do**
    if $\|\mathbf{u}_{i,arch} - \mathbf{u}_{i,k}\| \neq 0$ **then**
      Compute local maxima $\tilde{\mathbf{u}}^l_{i,arch}$ and $\tilde{\mathbf{u}}^l_i$ associated to $\mathbf{d}_{i,k}$ and $\mathbf{d}_{i,arch}$
      if $f_l(\mathbf{d}_{i,arch}, \tilde{\mathbf{u}}^l_i) \geq f^l_{arch}(\mathbf{d}_{i,arch}, \mathbf{u}^l_{i,arch})$ **then**
        replace $\mathbf{u}^l_{i,arch}$ with $\tilde{\mathbf{u}}^l_i$
        replace f^l_{arch} with $f_l(\mathbf{d}_{i,arch}, \tilde{\mathbf{u}}^l_i)$
      end if
      if $f_l(\mathbf{d}_{i,k}, \tilde{\mathbf{u}}^l_{i,arch}) \geq f_l(\mathbf{d}_{i,k}, \mathbf{u}^l_{i,k})$ **then**
        replace $\mathbf{u}^l_{i,k}$ with $\tilde{\mathbf{u}}^l_{i,arch}$
        replace $f_l(\mathbf{d}_{i,k}, \mathbf{u}^l_{i,k})$ with $f_l(\mathbf{d}_{i,k}, \tilde{\mathbf{u}}^l_{i,arch})$
      end if
    end if
  end for
end for

Algorithm 4 Validation

for all $l \in \{1, \ldots, m\}$ **do**
  $\Delta f_{best} = 1$
  while $\Delta f_{best} \neq 0$ **do**
    $j = \text{argmin}\, f_l \in A_k$
    run global optimization over $\bar{U}$ and compute new $\bar{f}_l$ with associated $\bar{\mathbf{u}}^l_{j,k}$
    if $\bar{f}_l > f_l$ **then**
      replace $\mathbf{u}^l_{j,k} \in A_k$ with $\bar{\mathbf{u}}^l_{j,k}$
      replace $f_l \in A_k$ with $\bar{f}_l$
      $\Delta f_{best} = \bar{f}_l - f_l$
    end if
  end while
end for

where $i, j = 1, \ldots, n$. In the same way, the correlation of the new point $\mathbf{x}^*$ at which we want to predict the response $\hat{y}(\mathbf{x}^*)$, with the design points can be expressed as

$$\mathbf{r} = [r_i] = \exp\left[-\sum_{l=1}^{d} \theta_l \|\mathbf{x}^*_l - \mathbf{x}_{il}\|^{p_l}\right] \tag{5}$$

Under the assumption that the regression function is a zero-th order polynomial, i.e. it is a $n \times 1$ vector of ones $\mathbf{1}$, the prediction $\hat{y}(\mathbf{x}^*)$ can be found to be

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{6}$$

where

$$\hat{\mu} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \tag{7}$$

One of the key benefits of Kriging is the provision of an estimated error in its predictions. The estimated mean squared error (MSE) for a Kriging model is

$$s^2(\mathbf{x}^*) = \hat{\sigma}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right] \tag{8}$$

where

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}_{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{n} \tag{9}$$

is the estimated variance. The availability of an estimate of the prediction error is a very convenient feature, as the surrogate update strategy can be based on it: If the MSE is above a certain threshold, the real model is called and the actual response $y^*(\mathbf{x}^*)$ is computed. The pair $(\mathbf{x}^*, y^*)$ is then added to the design sites and the surrogate model is trained again. It is worth noticing that the correlation model depends on two parameters θ and p. They can be found to be the ones that minimize the mean squared error between the predicted response $\hat{y}$ and the actual response y. Therefore an optimization is to be performed during the training phase.

III.B. Radial Basis Function Networks

A Radial Basis Function Network is a two layer neural network, where each hidden unit implements a radial basis activation function, whereas the output units implement a weighted sum of the hidden unit outputs. The input into a RBF network is nonlinear, whereas the output is linear. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which perceptron neural networks can only model by means of multiple intermediary layers.[10] Assuming the input of the network be modelled as a vector $\mathbf{x}$ of real numbers, the output of the network is a scalar function of the input vector, $\hat{y}(\mathbf{x})$:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{n} w_i h_i(\mathbf{x}) \tag{10}$$

where n is the number of units in the hidden layer, w_i is the weight of unit i in the linear output unit, and h is the activation function implementing a radial basis:

$$h_i(\mathbf{x}) = \rho\left(\|\mathbf{x} - \mathbf{c}_i\|\right) \tag{11}$$

where $\mathbf{c}_i$ is the center vector of unit i. The function ρ can be any radial basis function, but a usual choice is to take it to be Gaussian:

$$\rho\left(\|\mathbf{x} - \mathbf{c}_i\|\right) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma^2}\right] \tag{12}$$

The norm is typically taken to be the Euclidean distance, and the parameter σ is called width parameter or spread, and represents the width of an area in the input space to which each unit responds. In a RBF network there are three types of parameters that need to be chosen: the center vectors $\mathbf{c}_i$, the output weights w_i, and the width parameters σ. The phase in which these parameters are chosen, or computed, is called training. If we suppose to have a set of n design sites $(\mathbf{x}, \mathbf{y})$, in order for the network to perform well at interpolation the centers can be taken to be the known points $\mathbf{x}$, so that the activation functions take the form $h_{ij}(\mathbf{x}) = \rho\left(\|\mathbf{x}_j - \mathbf{x}_i\|\right)$, and the weights can be found by solving the equation

$$\begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y(\mathbf{x}_1) \\ y(\mathbf{x}_2) \\ \vdots \\ y(\mathbf{x}_n) \end{bmatrix} \tag{13}$$

hence

$$\mathbf{w} = \mathbf{H}^{-1}\mathbf{y} \tag{14}$$

Finally, the width parameter σ is taken to be equal to the dimension of the design space.

III.C. Construction of the Surrogate Models

Let us consider, without loss of generality, Problem 3 in the single-objective case:

$$\nu_{\max} = \min_{\mathbf{d} \in D} \max_{\mathbf{u} \in \bar{U}} f(\mathbf{d}, \mathbf{u}) \tag{15}$$

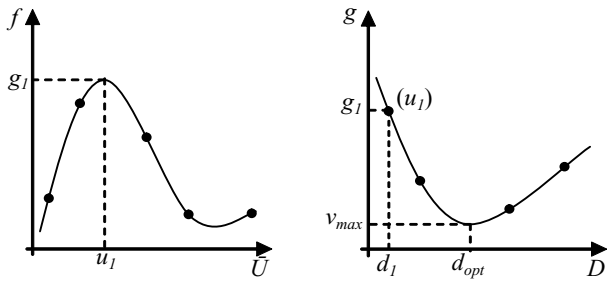
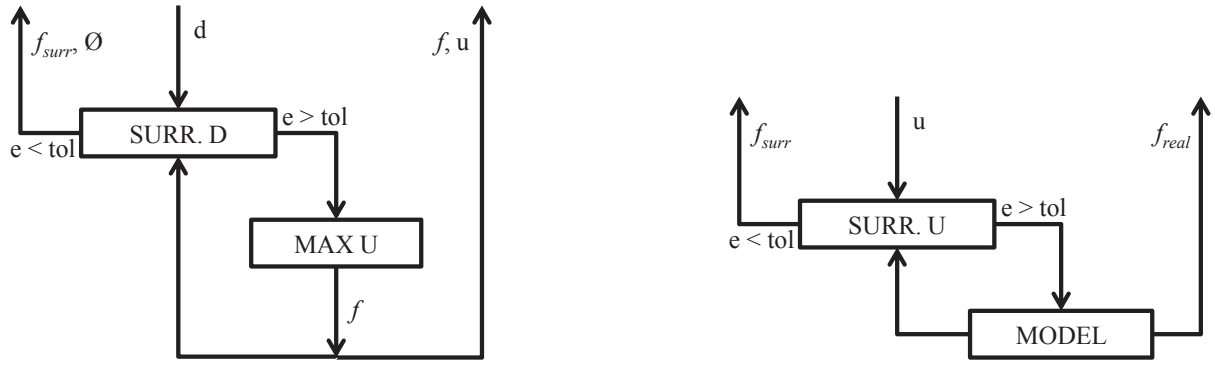


Figure 1. Conceptual example of the surrogate modelling strategy. The plots represent the surrogates in $\bar{U}$ (left) and D (right) spaces. The dots are the design sites. The surrogate in $\bar{U}$ is generated for a fixed d_1. Notice that $g_1 = \max_u f(d_1, u)$, and the pair (d_{opt}, ν_{max}) is not associated to any u.



(a) Surrogate model in the outer loop.

(b) Surrogate model in the inner loop.

Figure 2. Schematic of the surrogate modelling strategy.

This can also be written as

$$\nu_{\max} = \min_{\mathbf{d} \in D} g(\mathbf{u}) \tag{16}$$

where

$$g(\mathbf{u}) = \max_{\mathbf{u} \in \bar{U}} f(\mathbf{d}, \mathbf{u}) \tag{17}$$

i.e. it is the result of the inner loop of the optimization.

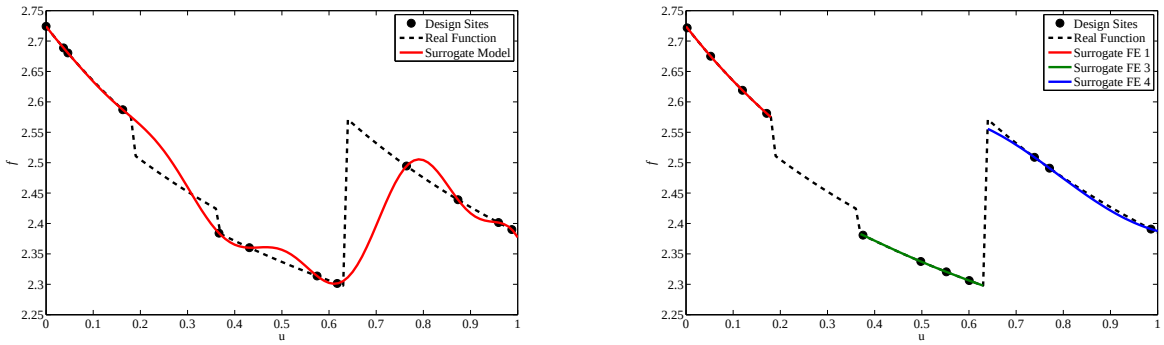
Two different surrogates are built in the D and $\bar{U}$ spaces respectively, as shown in Figure 1. The surrogate in the design space has $\mathbf{d}$ as design sites, and g as response, and is therefore constructed and updated in the outer loop of the optimization, i.e. the minimization over D. The surrogate in the uncertain space has $\mathbf{u}$ as design sites and f as response, and is therefore constructed and updated in the inner loop of the optimization, i.e. the maximization over $\bar{U}$ (Figure 2).

The aim of this approach is to have surrogate models that are as simple as possible during the optimization cycles, and each loop working only with the relevant parameters, i.e. the design vector $\mathbf{d}$ for the outer loop, and the uncertain vector $\mathbf{u}$ for the inner one. Another advantage of such approach is that the inner loop is called only if the surrogate $(\mathbf{d}, g)$ needs to be updated. If the accuracy of the surrogate $(\mathbf{d}, g)$ is above a certain threshold, only the outer loop is run, hence saving computational expense. It has to be noted that a possible consequence of this strategy is to have no uncertain vector as output. This is so because uncertain vectors are associated only to the responses g that come from the inner loop, i.e. the design sites of the surrogate $(\mathbf{d}, g)$. All the other points in the $(\mathbf{d}, g)$ curve have no uncertain vector associated to. In order to overcome this problem and get the optimal uncertain vector, after the minmax algorithm has found the optimal design vector $\mathbf{d}_{opt}$ a further maximization cycle is run over $\bar{U}$ in order to find the optimal uncertain vector $\mathbf{u}_{opt}$ associated to $\mathbf{d}_{opt}$ and $\nu_{\max}$.

Whereas one single surrogate approximates the whole design space, two strategies have been devised for the construction of the surrogate in $\bar{U}$. In one case, a single surrogate model approximates the whole uncertain space. This strategy will be referred to as global U surrogate, and an example of how it could look

like is shown in Figure 3(a). Such surrogate is globally good, but the quality of the approximation might be locally poor. Moreover, the surrogate dimension can be high, as it contains design sites for the whole $\bar{U}$ space. In order to increase the local quality of the approximation while decreasing the dimension of the surrogate, another strategy has been devised for the construction of the surrogate model in $\bar{U}$. Instead of having one global surrogate model, we construct many smaller surrogates, each one associated to one focal element. In this way the surrogates in $\bar{U}$ are locally good, and they contain only design sites relevant to a specific focal element. Indeed, as the optimization loop converges, the size of the search space becomes smaller, eventually converging to one focal element. Therefore only the surrogate associated to that focal element will be retained and used, increasing the quality of the approximation while keeping low the dimension of the surrogate. This second strategy will be referred to as FE-based surrogate, and an example of how it might look like is shown in Figure 3(b). Notice also that each focal element is a region generally smoother than the whole uncertain space. It is therefore easier to achieve a good quality approximation.

The surrogate models are built online: This means that the surrogate is trained progressively once a new agent is deployed into the optimization loop. As at least two design sites are needed in order for the surrogate to be trained, the first two agents that are deployed into the optimization loop are added to the list of design sites and the real model is called. From the third agent onwards, the surrogate model is used to estimate the response of the model. Then, if the update condition is met, the current agent is added to the list of design sites, the real model is called, and the surrogate is updated, i.e. is trained with the new list of design sites. The same approach is used for the surrogates in $\bar{U}$ and D spaces. The update condition depends on the surrogate model used. For the Kriging predictor, the surrogate is updated if the estimation error (MSE) is above a certain threshold, whereas for the RBF network the estimation error is not available, therefore a different approach is to be used. In this case we check the distance between the current agent and its closest neighbour included in the list of design sites. If this distance is above a certain threshold, the surrogate is updated. Such threshold is a user defined parameter, and is tunable, as it affects the quality of the approximation. Indeed, the lower the threshold, the more often the surrogate is updated, and the more accurate it approximates the true model, but also the more computationally expensive it is.



(a) Global U surrogate.

(b) FE-based surrogate.

Figure 3. Examples of Global U surrogate (a) and FE-based surrogate (b).

Moreover, in order to keep the size of the surrogates from becoming too large and expensive, a further check is implemented. When the number of design sites overcomes a certain threshold, the mean value of the responses of the design sites is computed. Then only the design sites which response is lower or equal to the mean value are retained, and the surrogate model is trained again. The idea behind this is that in a minimization, the responses with the lower value are the most interesting.

IV. Test Case

The techniques proposed in this paper were applied to the solution of a realistic case in which a space system made of a communications subsystem needs to be designed under uncertainty. The tests in this section aim at showing that the use of surrogate modelling in Evidence-Based Robust Optimization can provide sufficiently accurate results at a fraction of the computational cost.

IV.A. Communications System Model

The mass and power of the telemetry and telecommand system (TT&C) are computed starting from the link budget. The required communication link characteristics are the Bit Error Rate BER, the modulation, and the ground station antenna gain G_r. From BER and modulation one can compute the required energy per bit noise ration $EbNo$. The $EbNo$ plus the data rate are used to compute the Carrier to Noise Ration CN_{ratio}. The total amount of data transmitted is assumed to be $T_{data} = 10^3 B$, where B is the total amount of data coming from the command and data handling system to the TT&C one. Given the access time Δt_a, the required data rate R_t is calculated as follows:

$$R_t = 10 \log_{10} \left(\frac{T_{data}}{\Delta t_a - \Delta t_{aq}} \right) \tag{18}$$

where Δt_{aq} is the target acquisition time. Given the data rate and the bit to noise ratio, CN_{ratio} is simply

$$CN_{ratio} = EbNo + R_t \tag{19}$$

Now one can compute the Equivalent Isotropic Radiated Power $EIRP$ as follows:

$$EIRP = CN_{ratio} - G/T + L_{total} - k \tag{20}$$

where $k = 228.6$ dB, L_{total} is the total signal loss, and G/T is the receiving system performance. The total signal loss is computed adding up all the factors that lead to a loss of signal energy and an increase of noise. Here most of the losses or sources of noise have been modelled with simple equations or look-up tables. The free space losses F_{SL} are calculated from the distance from the ground station r_{GS} and the frequency of the transmitter f_T:

$$F_{SL} = 32.4 + 20 \log_{10} r_{GS} + 20 \log_{10} f_T \tag{21}$$

The polarization mismatch losses P_L can be computed from the Faraday rotation ϕ_f using the following relationship:

$$P_L = -20 \log_{10} (\cos \phi_f) \tag{22}$$

The atmospheric losses A_L are a function of the ground station altitude h_{GS}, and are collected in a look-up table (Table 1) and interpolated. The dependency of the atmospheric losses on the elevation angle is

Table 1. Atmospheric losses.

h_{GS} **(km)**	A_L **(dB)**
-2 to 2	0.04
2.1 to 6	0.025
6.1 to 10	0.008
10.1 to 14	0.004
14.1 to 18	0.001

modelled by introducing a simple sinusoidal function of the elevation angle e:

$$A_{LH} = \frac{A_L}{\sin e} \tag{23}$$

The rain absorption losses Ra_L are calculated by using data from.[11] The worst case losses for the feeder loss F_L, the antenna misalignment loss AM_L, and the implementation loss I_L are reported in Table 2 The total

Table 2. Worst case losses, in dB.

F_L	AM_L	I_L
2	0.5	2

loss L_{total} is obtained by summing up all the individual losses:

$$L_{total} = FS_L + F_L + AM_L + A_{LH} + P_L + Ra_L + I_L \tag{24}$$

The receiving system performance G/T is calculated as follows:

$$G/T = G_r - TS_{noise} \tag{25}$$

where G_r is the ground station antenna gain, and the total system noise TS_{noise} is determined as follows. First, the receiver noise figure RN_{fig} is computed:

$$RN_{fig} = AN_R + AMP_R + \frac{\left(10^{\frac{L_R}{10}} - 1\right) k_0 + 10^{\frac{L_R}{10}} \left(10^{\frac{F_R}{10}} - 1\right) k_0}{10^{\frac{G_R}{10}}} \tag{26}$$

where AN_R is the receiving antenna noise temperature, AMP_R is the receiving amplifier noise, L_R is the receiver cable loss, G_R is the receiver low noise amplifier gain, F_R is the receiver noise figure, and $k_0 = 290$. The transmitter noise temperature S_{temp} is:

$$S_{temp} = AN_T + AMP_T + \frac{\left(10^{\frac{L_T}{10}} - 1\right) k_0 + 10^{\frac{L_T}{10}} \left(10^{\frac{F_T}{10}} - 1\right) k_0}{10^{\frac{G_T}{10}}} \tag{27}$$

Here AN_T is the transmitter antenna noise temperature, AMP_T is the transmitter amplifier noise, L_T is the transmitter cable loss, G_T is the transmitter low noise amplifier gain, and F_T is the transmitter noise figure. The rain noise N_{rain} is then calculated:

$$N_{rain} = \left(1 - \frac{1}{10^{\frac{Ra_L}{10}}}\right) k_0 \tag{28}$$

The total system noise TS_{noise} then writes:

$$TS_{noise} = 10 \log_{10}(RN_{fig} + S_{temp} + N_{rain}) \tag{29}$$

Now, the required transmission power P_{TTC} onboard the spacecraft is defined as

$$P_{TTC} = CN_{ratio} - G/T + L_{total} - k - G_t \tag{30}$$

where G_t is the transmitter antenna gain. The total mass M_{TTC} of the communications system is given by the sum of the individual masses of antenna M_{ant}, amplifier M_{amp}, and casing M_{case}:

$$M_{TTC} = M_{ant} + M_{amp} + M_{case} \tag{31}$$

The spacecraft antenna type is chose on the basis of the required antenna gain G_t. It is known[12] that the best antenna for gain in the range 5-10 dB is the patch antenna, for gain in the range 10-20 dB is the horn antenna, and above 20 dB is the parabolic antenna. The mass of the antenna is computed as follows. The antenna characteristic length (the diameter of the conical section for parabolas, conical horns, and circular patches, and an equivalent diameter for pyramidal horns and square patches) is

$$D_{ant} = \frac{c}{\pi f_T} \sqrt{\frac{10^{\frac{G_T}{10}}}{\eta_{ant}}} \tag{32}$$

where c is the speed of light, and η_{ant} is the antenna efficiency. In case of patch antenna, the mass is

$$M_{ant,patch} = \pi \frac{D_{ant}^2}{4} \left(0.0015\rho_{diel} + 0.0005\rho_{copper}\right) \tag{33}$$

where $\rho_{diel} = 2000$ kg/m^3 and $\rho_{copper} = 8940$ kg/m^3 are respectively the average values of the dielectric material density and the copper density, and considering a total thickness of 2 mm, with 1.5 mm of dielectric material and 0.5 mm of copper. In case of horn antenna, the mass is

$$M_{ant,horn} = S_L \rho_A \tag{34}$$

where S_L is the lateral surface of the horn

$$S_L = \pi \frac{D_{ant}}{2} \sqrt{\frac{D_{ant}^2}{4} + L_{horn}^2} \tag{35}$$

Table 3. TT&C design space.

Parameter	Bounds
f_T (MHz)	[7e3 11e3]
Mod	[0 1]
T	[0 1]
η_{ant}	[0.5 0.95]
G_t (dB)	[5 20]
ρ_{CMR}	[0.1 0.3]

Table 4. Environmental parameters.

Parameter	Value
Faraday rotation (deg)	9
Ground station antenna gain (dB)	60
Distance from ground station (km)	1.5e6
Access time (s)	1000
Total amount of data (kb) (CDH output)	120e3
Bit Error Rate	1e-6
Ground station altitude (km)	0
Horizon elevation (deg)	30
Receiver	
Low noise amplifier gain (dB)	60
Cable loss (dB)	8
Amplifier noise (K)	400
Receiver Noise Figure	10
Low noise amplifier gain (dB)	40
Transmitter	
Cable loss (dB)	2
Amplifier noise (K)	400
Receiver Noise Figure	10
Antenna Noise Temperature (K)	60

where, from available data,[12] $L_{horn} = 2D_{ant}$ and the areal density $\rho_A = 15$ kg/m^2. In case of parabolic antenna, the mass is

$$M_{ant,dish} = \pi \frac{D_{ant}^2}{4} \rho_A \tag{36}$$

and the surface density ρ_A has a typical value of 10 kg/m^2. The mass of the amplifier is a function of the transmission power P_{TTC}, as well as the mass of the casing. Two types of amplifiers have been considered: TWTA and solid state. The amplifier mass is the result of the interpolation of existing pairs (M_{amp}, P_{TTC}). Finally, the casing mass is computed as a fraction of the amplifier mass

$$M_{case} = M_{amp} \rho_{CMR} \tag{37}$$

The models in this section are derived from Roddy,[11] Balanis,[12] and Wertz.[13]

The design parameters are: carrier frequency f_T, modulation Mod, amplifier type T, the antenna efficiency η_{ant}, the transmitter antenna gain G_t, and the amplifier casing mass fraction ρ_{CMR}. The design space for the TT&C system is summarized in Table 3. Other parameters involved in the model and deriving from environment, mission analysis, or other spacecraft subsystems are considered as fixed environmental parameters, and are collected in Table 4. Notice that the characteristics of the communications subsystem were not selected to reflect a real mission scenario but only to test the proposed methodology.

IV.B. Surrogate-Based Single-Objective Optimization

This subsection presents the results of the application of surrogate models to the computation of ν_{max}, solution of the min-max problem 3 with only one objective function, the mass of the TT&C system, as in Eq. 31. In this case the uncertain parameter chosen are the antenna efficiency η_{ant} and the amplifier casing mass fraction ρ_{CMR}. The *bba* structure is summarized in Table 5. Note that some intervals are overlapping.

The minimization over D is performed with a modified version of IDEA, called IDEAν.[6] The number of agents and function evaluations were set respectively to 5 and 300 for both the outer and inner loops. The Kriging predictor makes use of a zero-th order polynomial regression function and a Gaussian correlation function. The surrogate is updated when the root mean square error of the prediction is above a threshold of 0.1 kg. The RBF network makes use of a Gaussian activation function, and the width parameter σ is equal to 1. The surrogate is updated when the distance between the current point in the search space and its closest design site is greater than 1% of the size of the search space along each dimension. Table 6 summarizes the results of 100 simulations. The 'Mass' column contains the best value. The success rate indicates how many times the optimal mass is found, and, bracketed, how many times the optimal uncertain vector is found. The first row of Table 6 shows the results of the optimization problem when using the actual TT&C

Table 5. TT&C *bpa* structure for the single-objective optimization.

η_{ant}	Interval	[0.5 0.6]	[0.65 0.75]	[0.6 0.8]	[0.8 0.95]
	bpa	0.2	0.5	0.1	0.2
ρ_{CMR}	Interval	[0.1 0.2]	[0.25 0.3]	[0.1 0.3]	
	bpa	0.6	0.3	0.1	

Table 6. Results of the surrogate-based single-objective optimization.

Model	Design param.	Uncertain param.	Mass (kg)	Success rate (%)
TT&C model	[7000, 0, 1, 17]	[0.5, 0.3]	3	90(100)
Global U Krig.	[7000, 0, 1, 17]	[0.5, 0.3]	3	67(90)
FE-based Krig.	[7000, 0, 1, 17]	[0.5, 0.3]	3	62(93)
Global U RBFN	[7000, 0, 1, 17]	[0.5, 0.3]	3	32(75)
FE-based RBFN	[7000, 0, 1, 17]	[0.5, 0.3]	3	54(100)

Table 7. Average percentage of calls to the real model and to the surrogate models for the different surrogate modelling methods.

	Real model (%)	Surrogate model (%)
Global U Krig.	12	88
FE-based Krig.	17	83
Global U RBFN	55	45
FE-based RBFN	59	41

model. The optimal mass results to be equal to 3 kg, this value being both the minimum value and the most frequent one (i.e. the mode). The two numbers of the success rate are different because to the optimal uncertain vector may correspond a suboptimal value of the function. The rows from the second to the fifth of Table 6 show the results of the optimization problem when using the surrogate models in either of the two configurations we proposed in subsection III.C. It may be seen that, when using the Kriging as surrogate model, both Global U and FE-based strategies found the optimal uncertain vector about 90% of the times, whereas they found the optimal mass about 65% of the times. This discrepancy is due to the outer loop of the optimization, as for the results in the first row of Table 6. Besides, the error due to the surrogate in D contributes to achieving a lower success rate at finding the optimal mass. We believe that the advantage of the focal element-based surrogate over the global U surrogate will be even clearer in future tests with more complex systems. Also, according to the description of the surrogate modelling technique presented in Section III.C, Table 7 summarizes the average number of times, over 100 runs, that the real model and the surrogate model were called. It can be noted that, when using the Kriging predictor, the real TT&C model was called only about 15% of the times, whereas the surrogate was used for 85% of the function evaluations.

When using RBF networks as surrogate model, the advantage of using the focal element based surrogate, over the global one, is remarkable: the RBF network could always find the optimal uncertain vector with the FE-based surrogate, whereas it could find it only three times out of four with the Global U. This is due to the fact that, as discussed in III.C, the FE-based strategy makes the surrogate model approximate each single focal element, which is, in general, a region smoother than the whole uncertain space, therefore easier to be approximated. However, the RBF network proved to be less satisfactory than the Kriging predictor at finding the optimal mass. The quality of the results is lower despite the true TT&C model being used more often then the surrogate (see Table 7). This can be due to the less effective update strategy. Moreover, the computational time, in seconds, required by the RBF network is on average 10 times longer than the time required by the Kriging. This is due to the fact that the training process of a RBF networks is slower than the training of the Kriging predictor. Therefore the use of RBF networks as surrogate models in the case of an online construction of the surrogate such as the one implemented in this paper, where the training phase can be called many times, has a poorer effectiveness than the use of the Kriging predictor.

In addition, it is worth mentioning that the inner loop of the optimization, i.e. the maximization over the uncertain space, was called only 73% of the times, therefore with a reduction in the computational expense of 27%.

A further test has been implemented for the sake of verifying the dependency of the quality of the solution on the threshold set for the update of the surrogate. Using the Kriging predictor, and setting the threshold to 0.01 kg, i.e. one order of magnitude smaller than the threshold used in the previous tests, gave a success rate of about 63% for both the Global U and FE-based strategies, therefore without improvement, despite the real model being called 4500 times on average, i.e. 3 times more often.

IV.C. Surrogate-Based Multi-Objective Optimization

IV.C.1. Performance Metrics

In order to evaluate the performance of the proposed multi-objective optimization methods, two metrics can be used:[14]

$$M_{conv} = \frac{1}{N_p} \sum_{i=1}^{N_p} \min_{j \in M_p} 100 \left\| \frac{g_j - f_i}{g_j} \right\| \tag{38}$$

$$M_{spr} = \frac{1}{M_p} \sum_{i=1}^{M_p} \min_{j \in N_p} 100 \left\| \frac{f_j - g_i}{g_i} \right\| \tag{39}$$

where M_p is the number of elements, with objective vector g, in the true global Pareto front and N_p the number of elements, with objective vector f, in the Pareto front that a given algorithm yields. The two metrics measures two different things: M_{spr} is the sum, over all the elements in the global Pareto front, of the minimum distance of all the elements in the Pareto front N_p from the i-th element in the global Pareto front. M_{conv}, instead, is the sum, over all the elements in the Pareto front N_p, of the minimum distance of the elements in the global Pareto front from the i-th element in the Pareto front N_p. Therefore, if N_p is an accurate partial representation of the global Pareto front, M_{spr} will have a high value whereas M_{conv} will have a low value. If both metrics are high, then the Pareto front N_p is partial and poorly accurate. Given n repeated runs of a given algorithm, two performance indexes can be used:[14] $p_{conv} = P(M_{conv} < tol_{conv})$, i.e. the probability that the index M_{conv} achieves a value lower than a threshold tol_{conv}, and $p_{spr} = P(M_{spr} < tol_{spr})$, i.e. the probability that the index M_{spr} achieves a value lower than a threshold tol_{spr}. According to Vasile et al.,[14] 200 runs are sufficient to have a 95% confidence that the true values of p_{conv} and p_{spr} are within a $\pm 5\%$ interval containing their estimated value. The global front used in the test case was taken to be the Pareto front coming from the 200 simulations with $1 \cdot 10^6$ function evaluations.

IV.C.2. Real model

This subsection presents the results of the implementation of the multi-objective optimization algorithm proposed in section II. The objectives are mass and power of the TT&C system, as in Eq. 31 and 30. In this case the uncertain parameters chosen are the antenna efficiency η_{ant} and the gain of the ground station antenna G_r. The *bpa* structure is summarised in Table 5.

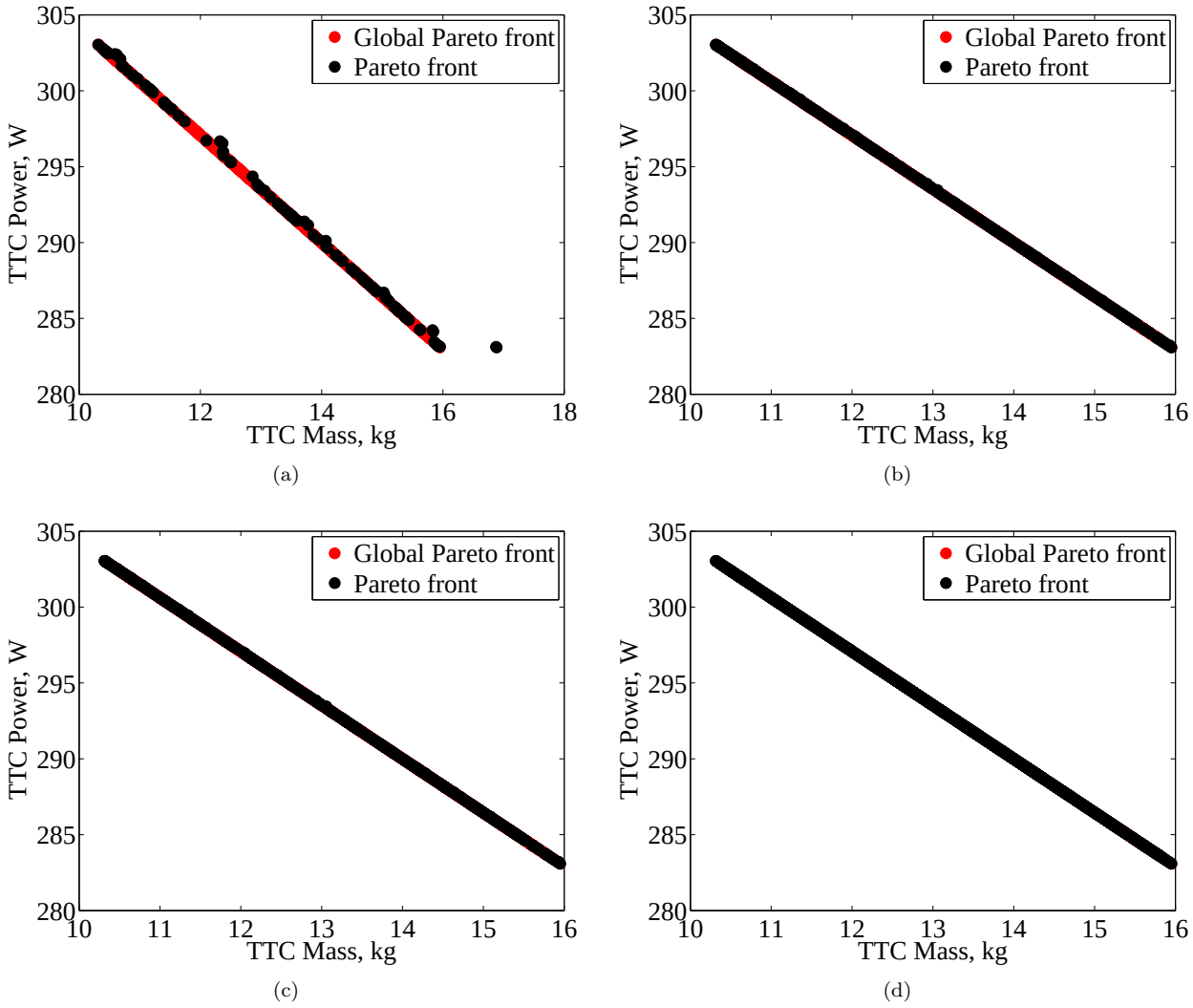


Figure 4. Pareto fronts of the TT&C model with (a) $1.5 \cdot 10^5$, (b) $3 \cdot 10^5$, (c) $5 \cdot 10^5$, and (d) $7.5 \cdot 10^5$ function evaluations.

The number of agents in the outer loop was set to 10 and the number of function evaluations was set to $1.5 \cdot 10^5$, $3 \cdot 10^5$, $5 \cdot 10^5$, $7.5 \cdot 10^5$, and $1 \cdot 10^6$. For the inner loop we used 5 agents and 300 function evaluations. The Pareto fronts solution of the multi-objective optimization problem are shown in Figure 4.

Table 8. TT&C *bpa* structure for the multi-objective optimization.

η_{ant}	Interval	[0.5 0.55]	[0.6 0.65]	[0.7 0.75]	[0.6 0.8]
	bpa	0.2	0.5	0.1	0.2
G_r	Interval	[5 15]	[10 25]	[15 60]	
	bpa	0.3	0.6	0.1	

Table 9 summarizes the performance metrics of the optimization algorithm at several numbers of function evaluations, taking the simulations with $1 \cdot 10^6$ function evaluations as the global Pareto front. The columns 'M_{conv}' and 'M_{spr}' contain the mean value and, in brackets, the standard deviation of M_{conv} and M_{spr} respectively. The thresholds tol_{conv} and tol_{spr} where set equal to 0.5 and 5 respectively. As expected, increasing the number of function evaluations the representation of the global Pareto front becomes more complete and accurate, i.e. the values of M_{conv} and M_{spr} decrease.

Table 9. Performance of the multi-objective optimization algorithm.

Func. Evals.	No. of points in the PF	M_{conv}	M_{spr}	p_{conv}	p_{spr}
$1.5 \cdot 10^5$	153	21.28(21.66)	47.25(48.4)	15.5%	2%
$3 \cdot 10^5$	636	3.95(6.58)	12.2(10.26)	46.5%	24%
$5 \cdot 10^5$	678	3.75(6.36)	11.73(9.9)	47.5%	26.5%
$7.5 \cdot 10^5$	2570	0.06(0.09)	1.08(0.84)	99.5%	99%
$1 \cdot 10^6$	2912	-	-	-	-

IV.C.3. Surrogate

This subsection presents the results of the application of surrogate models to the solution of the multi-objective optimization problem discussed in the previous subsection. Note that for this case we have used the same approach we used for the single-objective optimization problem. However, for the multi-objective optimization case, we have used only the Kriging predictor as surrogate model. This is because the results obtained by using the RBF network were not such to justify their use in a more computationally demanding problem (see the discussion at the end of subsection IV.B). The Kriging predictor makes use of a zero-th order polynomial regression function and a Gaussian correlation function. The surrogate is updated when the root mean square error of the prediction is above a threshold of 0.1 kg for the mass and 0.1 W for the power. As for the test case with the real model, the number of agents in the outer loop was set to 10 and the number of function evaluations was set to $1.5 \cdot 10^5$, $3 \cdot 10^5$, $5 \cdot 10^5$, $7.5 \cdot 10^5$, and $1 \cdot 10^6$. For the inner loop we used 5 agents and 300 function evaluations. Tables 10 and 11 summarize the performance metrics for global approach and the focal element one respectively. From Tables 10 and 11 it can be seen that

Table 10. Performance of the Global U surrogate multi-objective optimization algorithm.

Func. Evals.	M_{conv}	M_{spr}	p_{conv}	p_{spr}
$1.5 \cdot 10^5$	29.12(25.96)	69.99(84.92)	4.5%	0%
$3 \cdot 10^5$	14.05(51.92)	51.11(325.30)	20.5%	8%
$5 \cdot 10^5$	4.53(12.64)	11.91(30.39)	42%	43.5%
$7.5 \cdot 10^5$	2.91(11.67)	9.16(28.38)	57.5%	55%
$1 \cdot 10^6$	1.77(8.61)	7.14(26.90)	72.5%	72%

Table 11. Performance of the FE-based surrogate multi-objective optimization algorithm.

Func. Evals.	M_{conv}	M_{spr}	p_{conv}	p_{spr}
$1.5 \cdot 10^5$	32.35(28.26)	82.93(107.60)	6%	1%
$3 \cdot 10^5$	13.55(19.23)	33.47(52.23)	16%	6.5%
$5 \cdot 10^5$	6.54(15.58)	18.67(46.73)	39%	38%
$7.5 \cdot 10^5$	5.91(14.95)	16.17(45.57)	47%	57.5%
$1 \cdot 10^6$	4.87(14.24)	14.27(45.26)	62.5%	67.5%

the quality of the approximation increases with the number of function evaluations, as the values of M_{conv} and M_{spr} decreases. Overall, the Global U surrogate performs better than the FE-based one in terms of convergence and spreading at high numbers of function evaluations, whereas the performance is similar for less function evaluations. Moreover, it can be noted that the performance indexes p_{conv} and p_{spr} show that the two strategies are comparable. Tables 12 and 13 show the saving in computational expense obtained by using the surrogate model. Indeed, it can be seen that the real TT&C model was used for only about 40% of the total number of function evaluations, whereas the surrogate was used for about 60% of the times. It can be also noticed that the calls to the real model slightly decrease while increasing the total number of function evaluations. This is due to the fact that the quality of the surrogate model increases at higher numbers of function evaluations, and therefore the update condition is satisfied less often, i.e. the surrogate does not need to be updated. In addition, the inner loop of the optimization, i.e. the maximization in the uncertain space, is called on average 90% of the times. There is therefore a reduction of 10% in the computational

Table 12. Average percentage of calls to the real model and to the surrogate models at increasing numbers of function evaluations and using the Global U surrogate.

	Real model (%)	Surrogate model (%)
$1.5 \cdot 10^5$	46	54
$3 \cdot 10^5$	45	55
$5 \cdot 10^5$	43	57
$7.5 \cdot 10^5$	43	57
$1 \cdot 10^6$	41	59

Table 13. Average percentage of calls to the real model and to the surrogate models at increasing numbers of function evaluations and using the FE-based surrogate.

	Real model	Surrogate model
$1.5 \cdot 10^5$	43	57
$3 \cdot 10^5$	41	59
$5 \cdot 10^5$	41	59
$7.5 \cdot 10^5$	41	59
$1 \cdot 10^6$	40	60

expense.

The Pareto fronts obtained increasing the number of function evaluations and by using the Global U surrogate are shown in Figures 5(a) - 5(e), and in Figures 6(a) - 6(e) by using the FE-based surrogate. It can be seen that, increasing the function evaluations, the surrogate Pareto fronts improves, as shown also in Table 10 and Table 11. In particular it can be noticed that the number of non-dominated solutions increases and also that the such solutions approximate better the reference Pareto front, i.e. the convergence increases and the spreading decreases.

V. Conclusion

In this paper we presented a formulation for an evidence-based robust multi-objective optimization and the use of surrogate modelling to reduce the computational cost of its solution. We proposed an algorithm that implements special heuristics to increase the probability to accurately solve the min-max problem deriving from the 'worst case scenario' design with multiple objectives. Moreover, we devised a focal element-based technique to construct the surrogate model. The approach was applied to the worst case design of a satellite communications system with a number of uncertain parameters. Our tests showed that the use of surrogate models, and in particular of the focal element based approach, yields good results. The Kriging predictor as surrogate model proved to be a satisfying at approximating the real model and being computationally effective. On the other hand, the RBF network gave worse results in both the quality of the approximation and the computational demand. Overall, the techniques proposed in this paper gave satisfactory results for both the single-objective optimization and the multi-objective optimization. In particular, the focal element-based surrogate modelling proved to be a valid technique that needs to be further investigated and developed.

Acknowledgments

This work is partially supported through an ESA/NPI grant 'Evidence-based Robust Design Optimization'.

References

[1]Vasile, M., Minisci, E., and Wijnands, Q., "Approximated Computation of Belief Functions for Robust Design Optimization," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, April 2012.

[2]Oberkampf, W. L. and Helton, J. C., "Investigation of Evidence Theory for Engineering Applications," *43rd*

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 2002.

[3]Vasile, M., "Robust Mission Design Through Evidence Theory and Multiagent Collaborative Search," *Ann. N.Y. Acad. Sci.*, Vol. 1065, December 2005, pp. 152–173.

[4]Croisard, N., Vasile, M., Kemble, S., and Radice, G., "Preliminary Space Mission Design Under Uncertainty," *Acta Astronautica*, Vol. 66, No. 5-6, March-April 2010, pp. 654–664.

[5]Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, 1976.

[6]Vasile, M., Minisci, E., and Locatelli, M., "An Inflationary Differential Evolution Algorithm for Space Trajectory Optimization," *Trans. Evol. Comp*, Vol. 15, No. 2, April 2011, pp. 267–281.

[7]Zuiani, F. and Vasile, M., "Multi-agent Collaborative Search with Tchebycheff Decomposition and Monotonic Basin Hopping Steps," *5th International Conference on Bioinspired Optimization Methods and their Applications*, May 2012.

[8]Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., "Surrogate-Based Analysis and Optimization," *Prog. Aerosp. Sci.*, Vol. 41, No. 1, January 2005, pp. 1–28.

[9]Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, December 2001, pp. 345–383.

[10]Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 1994.

[11]Roddy, D., *Satellite Communications*, McGraw-Hill, 3rd ed., 2001.

[12]Balanis, C. A., *Antenna Theory: Analysis and Design*, Wiley-Interscience, Hoboken, NJ, 3rd ed., 2005.

[13]Wertz, J. R. and Larson, W. J., *Space Mission Analysis and Design*, Microcosm Press, Hawthorne, CA, 3rd ed., 1999.

[14]Vasile, M. and Zuiani, F., "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proc. IMechE, Part G: J. Aerospace Engineering*, Vol. 225, No. 11, September 2011, pp. 1211–1227.

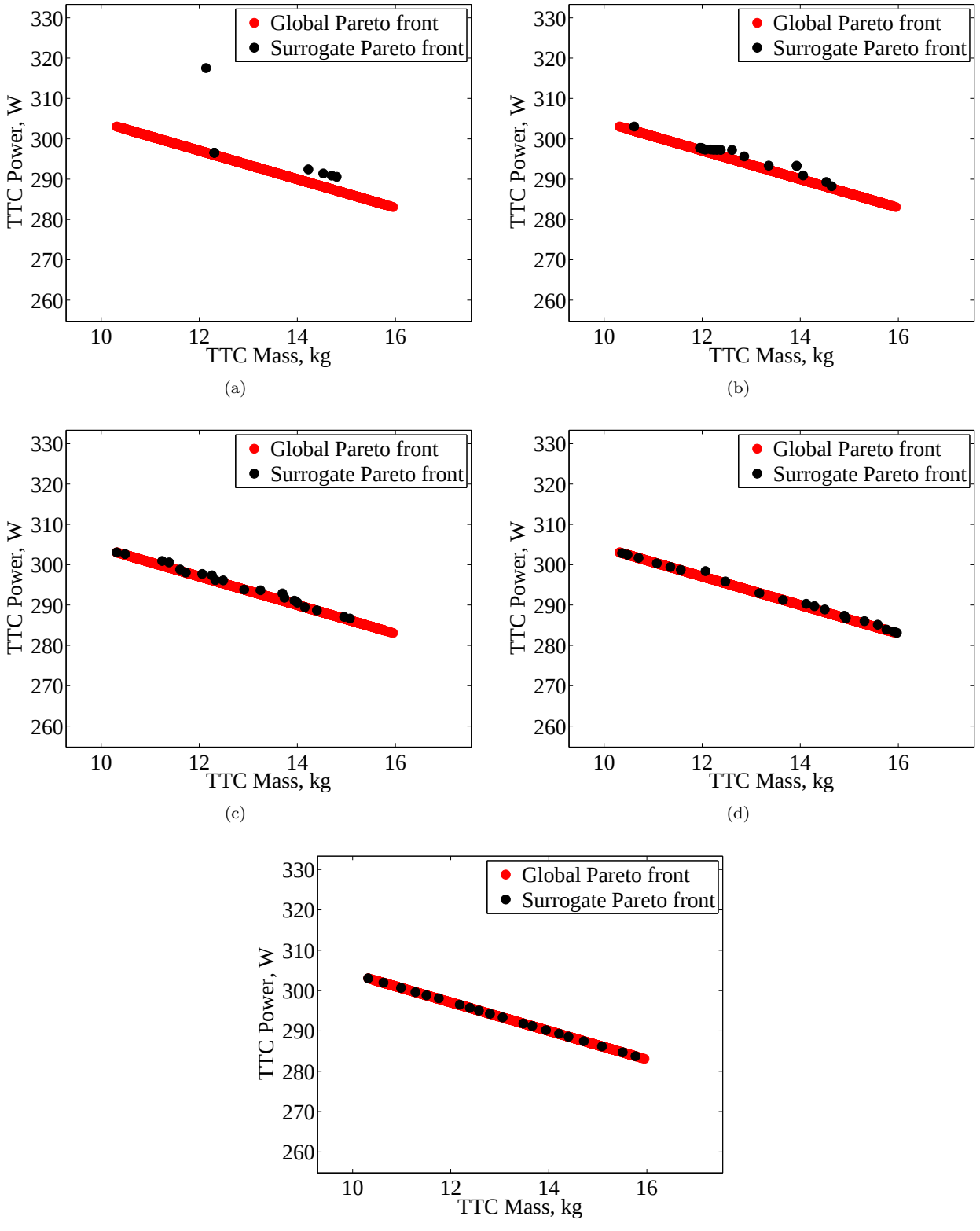


Figure 5. Pareto fronts of the Global U surrogate algorithm with (a) $1.5 \cdot 10^5$, (b) $3 \cdot 10^5$, (c) $5 \cdot 10^5$, (d) $7.5 \cdot 10^5$, and (e) $1 \cdot 10^6$ function evaluations.

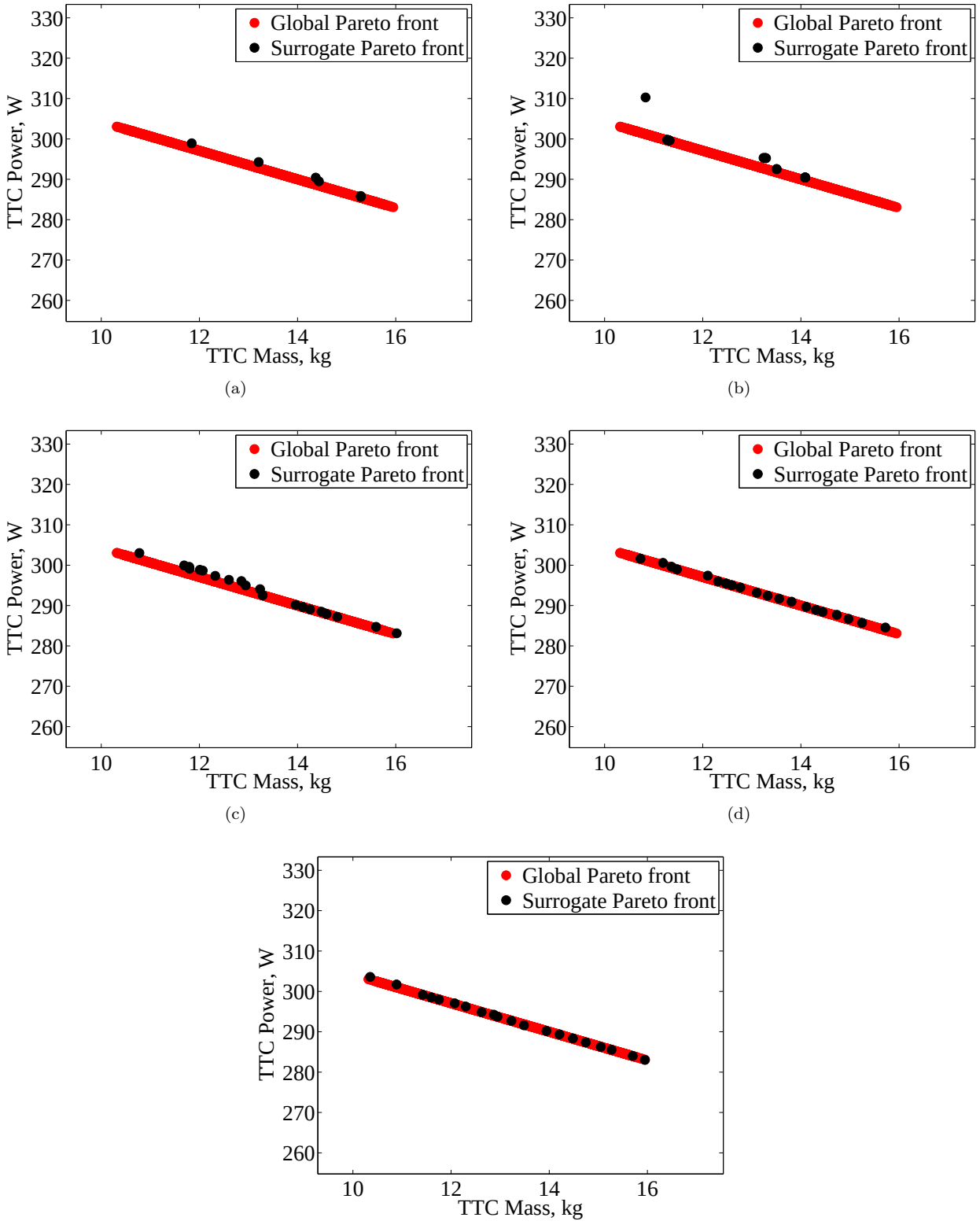


Figure 6. Pareto fronts of the FE-based surrogate algorithm with (a) $1.5 \cdot 10^5$, (b) $3 \cdot 10^5$, (c) $5 \cdot 10^5$, (d) $7.5 \cdot 10^5$, and (e) $1 \cdot 10^6$ function evaluations.